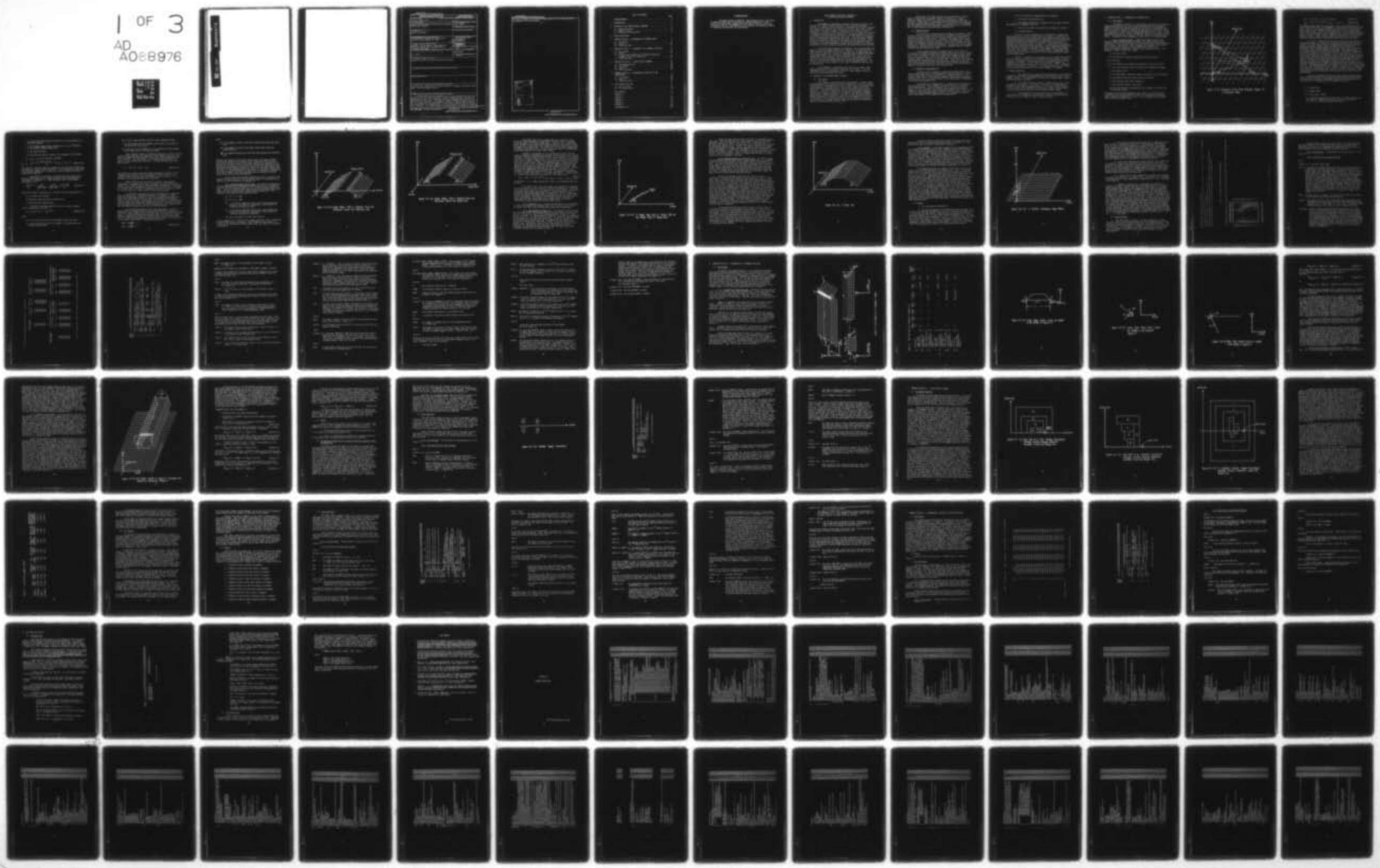
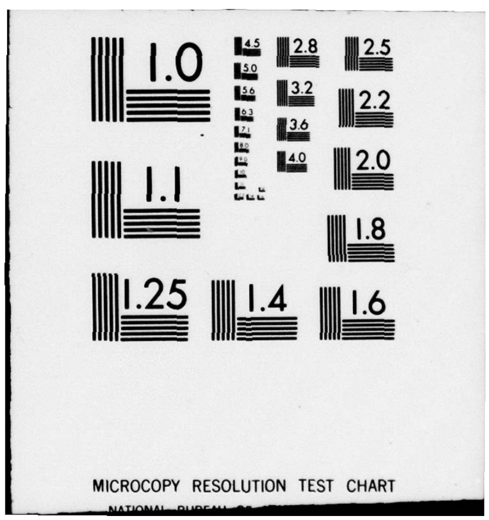
AD-A068 976 ARMY MATERIEL SYSTEMS ANALYSIS ACTIVITY ABERDEEN PROV--ETC F/G 19/3
LIGHTLY ARMORED STRUCTURE VULNERABILITY ESTIMATION METHODOLOGY --ETC(U)
JAN 79 C J LAPOINTE
UNCLASSIFIED AMSAA-TR-254 NL

1 OF 3
AD
A068976




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3
MICROCOPY RESOLUTION TEST CHART

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 6 Technical Report 254	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER 9
4. TITLE (and Subtitle) Lightly Armored Structure Vulnerability Estimation Methodology (LASVEM).		5. TYPE OF REPORT & PERIOD COVERED Final Report.
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) 10 Claude J. Lapointe		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS US Army Materiel Systems Analysis Activity Aberdeen Proving Ground, MD 21005		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS DA Project No. 16 1R765706M541
11. CONTROLLING OFFICE NAME AND ADDRESS Commander, US Army Materiel Development and Readiness Command, 5001 Eisenhower Avenue Alexandria, VA 22333		12. REPORT DATE 11 January 1979
		13. NUMBER OF PAGES 202 12 207p.
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) 14 AMSAA-TR-254		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

survivability; vulnerability; overhead protection; fragment; perforation; Thor; armor; armored personnel carrier

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A computer model has been developed to aid in the evaluation of the vulnerability of lightly armored structures, such as armored personnel carriers, to fragmentation, such as is produced by indirectly delivered high explosive (not ICM) field artillery shell. The program, coded in FORTRAN Extended, version 4, for CDC systems operating under NOS/BE 1 and SCOPE 2, depends upon the concept that vulnerability to fragmentation is directly related to the number of perforations sustained. This report presents the computational techniques and supporting

over

JRB

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

403 910

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

rationale, together with a program listing, details of the input data decks, and examples of input and output.

ACCESSION for		
NTIS	White Section	☑
DDC	Buff Section	☐
UNANNOUNCED		☐
JUSTIFICATION		
BY		
DISTRIBUTION/AVAILABILITY CODES		
Dist.	AVAIL. and/or	SPECIAL
A		

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

TABLE OF CONTENTS

ACKNOWLEDGEMENT

The author wishes to thank Mr. John Blomquist for his assistance in modifying Roger Snow's effectiveness computer program, Mr. Larry Kravitz for many helpful suggestions which resulted from his use of the methodology, and Ms. Cinda Roberts for her patient and persevering efforts in preparing the manuscript for publication.

LIGHTLY ARMORED STRUCTURE VULNERABILITY
ESTIMATION METHODOLOGY (LASVEM)

1. INTRODUCTION

The USAMSAA has developed a computer model which assists in the evaluation of lightly armored structures, such as armored personnel carriers, by computing various measures of vulnerability of those structures when subjected to indirect fire, high explosive (not ICM) field artillery projectiles. The basic concept is that the primary function of such a structure, with respect to field artillery fire, is to prevent the intrusion of fragments into an enclosure; therefore, a fundamental measure of such a structure's vulnerability to artillery fire is the number of perforations which it sustains when subjected to such fires. Such a quantity, however, may not be especially useful to decision makers at various levels. Hence, a family of statistical parameters, all based upon the expected number of perforations, has been developed to address the vulnerability of a group of structures to various volumes of tactically delivered field artillery fires.

The program was originally developed for use on the USA Ballistic Research Laboratory (USABRL) BRLESC I/II computers. The characteristics of those computers resulted in restrictions on the size of certain matrices used in the program and in the use of a magnetic tape oriented file system. As documented herein, the program has been modified to run on the new USABRL CYBER 173/76 computer. The file system is now random access, but the size of the above mentioned matrices has not been increased in the desire to minimize the changes made to an already debugged program. This is the source of certain remarks in this report concerning restraints due to machine size.

This document is intended to serve as a user's guide so that other organizations may achieve a capability in this area. It is also intended to serve as a programmer's guide for those changes to the program which may be necessary or desired.

2. OVERVIEW OF THE COMPUTATIONAL TECHNIQUE

2.1 The Program.

The program is divided into four main sections. Section I uses the methodology of Thor reports 47 and 70 (references 1 and 2) to produce a description of the damage sustained by a single plate or plate pair from the detonation of a projectile at a fixed point in space with a fixed angle of fall and terminal velocity, for all locations of the plate or plate pair. Section II combines the descriptions of damage sustained by several plates (output of Section I) to obtain a description of damage sustained by a complete structure. Section III uses the descriptions of damage sustained by a complete structure from the detonation of a projectile to obtain various statistical measures of the damage sustained by specified arrays of target structures when these arrays are subjected to tactically delivered field artillery fire.

Whereas the first three sections of the program deal with the number of perforations sustained, Section IV assesses, in a rather simplistic manner, the effect of a perforation upon people within the enclosure. Additionally, it determines, without reference to the detailed and time consuming computations of Section I, whether any of the fragments have the capability of ever penetrating (except by direct hit or very near miss of the projectile) the plate under consideration.

2.2 Damage Criteria.

The function of a lightly armored structure with respect to field artillery fire is to prevent the intrusion of fragments into an enclosure. (Protection against direct hits and very near misses seems impossible to achieve with current light armors.) Therefore, the damage criteria used herein are always of the form, "sustain at least N perforations", and the probability statements obtained are always of the form, "probability that M targets each sustain at least N perforations". For brevity, this is often expressed as, "probability that M vehicles are damaged", where the damage criterion has been specified as above.

A minor problem arises with damage criteria of this form if one wishes to differentiate between the damage (which may be major) caused by individual fragments and the truly catastrophic damage caused by direct hits or very near misses. Within this model, the problem has been eliminated by forcing a direct hit or very near miss to produce a very large expected number of perforations (currently 10^4). Thus, direct hits and very near misses are addressed within the established framework, and yet are distinguished merely by using a criterion "sustain at least N perforations", where N is large but still small with respect to 10^4.

2.3 Measures of Vulnerability.

Sections I and II produce an expected perforations matrix for a single target subjected to the fragmentation of one projectile at a fixed point in space with a fixed angle of fall and terminal velocity. Each element of this matrix is the expected number of perforations sustained by the target when it is located at a point in the ground plane corresponding to the matrix coordinates, (i,j).

The probability of damage matrix is obtained from the expected perforations matrix. With a damage criterion of the form, "sustain at least N perforations", and the assumption that a Poisson distribution applies for each location of the target, each element of this matrix is the probability that the target sustains damage at that location. There is a one-to-one correspondence between elements of the expected perforations matrix and elements of the probability of damage matrix.

From the probability of damage matrix are obtained:

a. The lethal area (reference 3).

b. The average probability of damage within the area over which the probability of damage is non-zero.

c. The area over which the probability of damage is non-zero.

d. A damage function.

Section III uses the methodology of reference 4 to address the vulnerability of an array of targets to various volumes of artillery fire. Using damage functions (one for each HOB in the HOB distribution being considered) obtained from the probability of damage matrices, this section of the program computes the probability that each target of the array is damaged for each volume of fire, for each HOB. The probabilities can be averaged over a specified HOB distribution. Using the average probabilities (over the HOB distribution) that each target sustain damage, the probabilities that various numbers of targets sustain damage, for each specified volley size, are computed. Also, an optimum HOB can be determined, which is then treated as an HOB distribution of zero width to obtain the quantities discussed above.

Section IV identifies those fragments of the fragment distribution which are capable of perforating a specified plate assuming no air drag. For each such fragment, the mass and velocity of the most massive residual fragment are computed. For each most massive residual fragment, the probability of incapacitating a soldier, given that he is hit, is computed using the methodology of reference 5. The average probability of incapacitation over the distribution of perforating fragments is computed.

3. MAJOR ASSUMPTIONS

a. The perforation of the specified armor plates by the specified fragments is described by the methodology of Thor reports 47 and 70 (references 1 and 2). This assumption is fairly easy to change should a better methodology become available.

b. The lightly armored structure is described by a set of flat, rectangular armor plates. If a plate is horizontal, its edges must be parallel to the range and deflection axes. If a plate is non-horizontal, it can deviate from a congruent, but horizontal, plate by not more than a rotation downward about any one of the horizontal plate's edges.

c. The probability of sustaining at least N perforations is a good measure of the degree to which indirect artillery fire with HE projectiles is able to defeat a lightly armored structure.

4. PROGRAM SECTION I - VULNERABILITY OF ARMOR PLATES

4.1 The Program.

The purpose of Section I is to determine, for one set of projectile fragmentation data and one set of projectile coordinates, the expected number of perforations sustained by one armor plate of given material and geometry for every location of interest of that plate in a horizontal plane. This information is placed in the expected perforations matrix file. During the computation, various outputs are generated, which are of interest in the themselves and which may, in fact, be the only reason for processing certain plates. Conceptually, the methodology of this section is straightforward.

4.1.1 Horizontal Plates. Rather than compute directly the expected number of perforations sustained by the target plate for every location of interest, we consider an infinite plate of the specified material and thickness. We establish a grid of square cells thereon, and fix the grid's location with respect to the projectile by defining its burst point (range, deflection) in terms of grid columns and rows. The geometry is shown in Figure 4-1, where:

Z is the burst point.

OZ is the vertical separation between burst point and plate.

O'Z is the HOB.

ZQ is the tangent to the projectile trajectory at the burst point.

P is the point currently being considered for fragmentation damage.

ω is the angle of fall.

Ψ is the supplement of the polar angle to the point P from the point Z.

$\hat{n}$ is an outward drawn unit vector normal to the plate.

ζ is the obliquity angle and is equal to Ψ.

ϕ is the azimuthal angle to the point P.

Θ is the angle between the projectile axis (assumed to coincide with ZQ) and the line ZP.

The equation for the angle between any two vectors in a spherical polar coordinate system yields equation 4-1 which, in a Cartesian coordinate system, reduces to equation 4-2, which is used to determine the fragment spray within which point P lies.

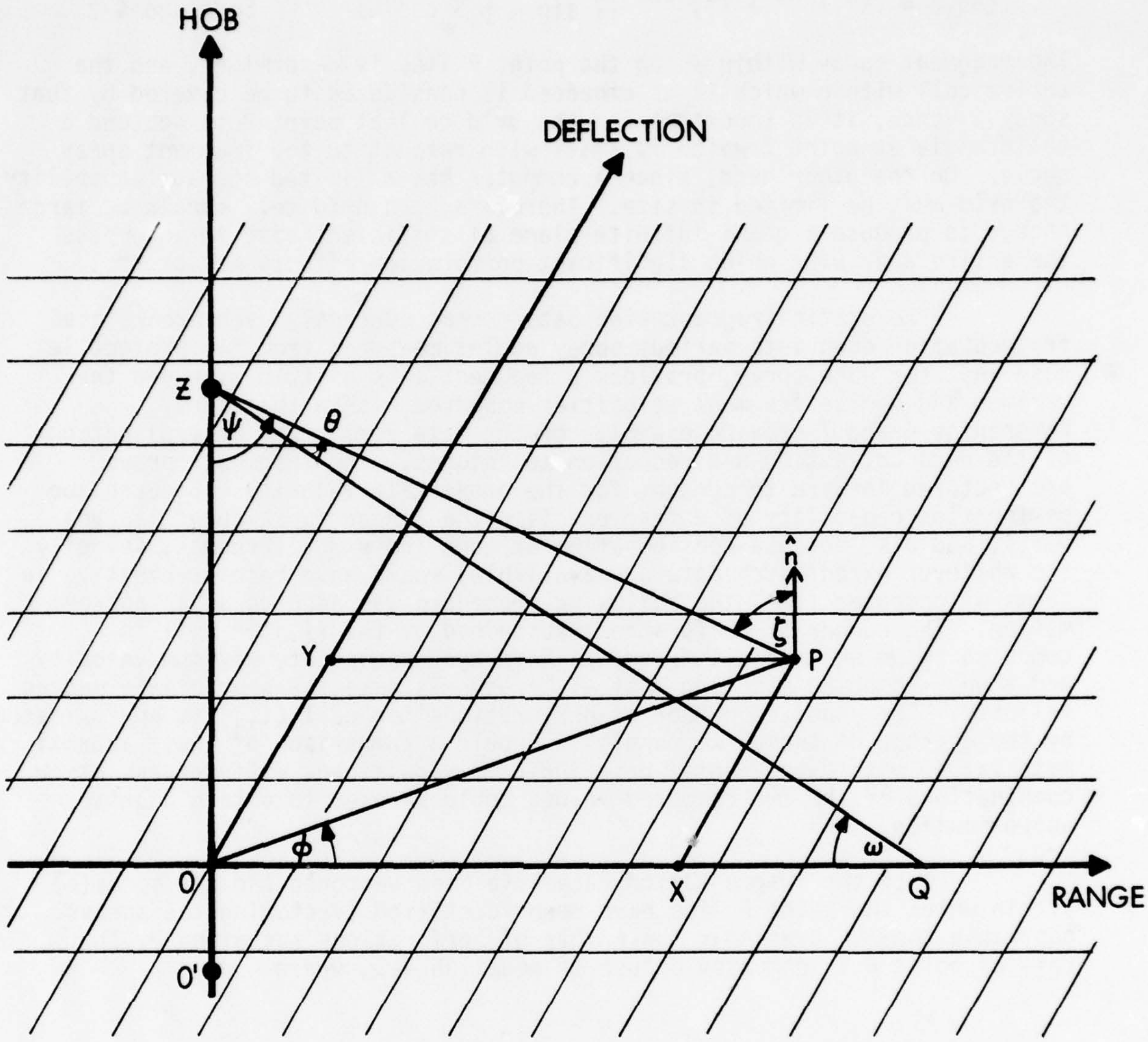


Figure 4.1 (U) Geometry of the Plate-Projectile System for a Horizontal Plate.

$$\cos\theta = \cos\psi \sin\omega + \sin\Psi \cos\omega \cos\phi \qquad \text{Equation 4-1}$$

$$\cos\theta = (X^2 + Y^2 + Z^2)^{-1/2} (Z \sin\omega + X \cos\omega) \qquad \text{Equation 4-2}$$

The fragment spray within which the point P lies is determined, and the entire cell within which it is embedded is considered to be covered by that spray. Hence, it is important for the grid cell at point P to subtend a solid angle at point Z which is small with respect to the fragment spray angle. On the other hand, since a computer has a limited storage capability, the grid must be limited in size. Therefore, the grid cell should be large enough to produce a quasi-infinite plane of sufficient size to encompass the entire area over which significant perforation effects may occur.

The static fragmentation data format currently used breaks the fragmentation down into various spray angles measured from the projectile nose and, for each spray, provides a fragment mass distribution and the maximum and median fragment velocities observed within that spray. References 6 and 7 provide examples of the data format and an explanation of the data collection and reduction techniques. The fragment sprays are vectored forward to account for the projectile velocity. Because the perforation capability of a fragment is not a linear function of its velocity, and also because consideration of each individual fragment velocity (to whatever extent such data are available) would have been prohibitive in terms of computer time, the following procedure was adopted as an approximation. The number of perforations sustained by the $(i,j)^{th}$ cell is computed assuming that all fragments have the appropriate maximum velocity and then recomputed assuming that all fragments have the appropriate median velocity. The expected number of perforations for cell (i,j) is approximated by the average of these two results. Should a comparison of these approximate values with experimental data indicate significant differences, other combinations of the two computed values could be used to obtain a better approximation.

Once the fragmentation data have been vectored and the spray(s) within which the point P lies have been identified (vectoring the sprays can cause them to overlap), the number of perforations sustained by the cell at point P is computed by use of equation 4-3, where

M is fragment mass.

V is fragment speed.

Q is either mass or speed.

r is a subscript identifying residual mass or residual speed of the most massive fragment particle resulting from a perforation. $M_r \leq 0$ or $V_r \leq 0$ means no perforation occurs.

s is a subscript identifying striking mass or striking speed of an impinging fragment.

K is the fragment shape factor (reference 7); $K = m_s\ A^{-3/2}$ where A is the average fragment impact area.

t is the plate thickness.

ζ is the obliquity angle at impact of the fragment with the plate.

C_Q, α_Q, β_Q, γ_Q, λ_Q are empirical constants.

$$Q_r = Q_s - 10^{C_Q}\ t^{\alpha_Q}\ K^{-(2/3)\alpha_Q}\ M_s^{(2/3)\alpha_Q + \beta_Q}\ (\sec\zeta)^{\gamma_Q}\ V_s^{\lambda_Q} \qquad \text{Equation 4-3}$$

This equation, from Thor Report 47 (reference 1), yields the residual mass and speed of a fragment after perforation. In the situation at hand, that of a horizontal plate, cos ζ is just cos Ψ which is merely the ratio of length $\overline{ZO}$ to length $\overline{ZP}$.

Thor Report 70 (reference 2) provides equations which can be used to extend Equation 4-3 to some aluminum alloys not considered in Thor Report 47 (reference 1). These equations, after some algebraic manipulation yield

$$\frac{Q_r(AL)}{Q_s} = 1 - \frac{k_Q(AL)}{k_Q(AL\ 2024)} + \frac{k_Q(AL)}{k_Q(AL\ 2024)}\ \frac{Q_r(AL\ 2024)}{Q_s} \qquad \text{Equation 4-4}$$

where the notation is the same as in equation 4-3 with the addition of:

k_Q is an empirical constant.

AL references some particular aluminum alloy.

AL 2024 references 2024 aluminum.

Reference 6 provides equation 4-5 which is used to compute velocity decay due to air drag:

$$V_s = V_o \exp\left(-AA \cdot R \cdot M_s^{-(1/3)}\right) \qquad \text{Equation 4-5}$$

where

V_s is the velocity with which the fragment strikes the plate

V_o is the velocity with which the fragment is projected from the fragment source point.

AA is the air drag constant provided in the fragmentation data.

R is the distance from the fragment source point to the center of the plate cell being struck.

M_s is the mass of the fragment as it is projected from the fragment source point and as it strikes the plate.

When fragments capable of perforating the plate at point P (M_r and V_r both greater than zero) are identified, their densities per unit solid angle are summed and then multiplied by the solid angle subtended by the cell at point P. The solid angle, Ω, is computed using equation 4-6, where A is the surface area of the cell, ζ is the obliquity angle, and Ω must be small.

$$\Omega = A \cdot (X^2 + Y^2 + Z^2)^{-1} \cdot \cos \zeta \qquad \text{Equation 4-6}$$

The preceding is done for both the maximum and median velocities of the fragments striking point P and the average of the two results is the expected number of perforations sustained by the cell at point P. This procedure is performed for each cell in the grid.

When a direct hit or very near miss occurs, the preceding computations are bypassed and a very large number of perforations (currently 10^4) is inserted into the matrix elements corresponding to the quasi-infinite plate cells sustaining a direct hit or very near miss. The direct hit problem, then, boils down to identifying the cells sustaining a direct hit or very near miss. This is done in the following manner.

A sphere with radius equal to the blast parameter is generated about the burst point. The upper hemisphere is replaced by a right circular cylinder whose directrix is the circle formed by the hemisphere edge. The intersection of the resulting solid figure with the quasi-infinite plate is a circle. On the quasi-infinite plate is constructed the smallest square which completely encompasses all cells, any parts of which were enclosed by the circle. The cells encompassed by this square are the ones considered to sustain direct hits or very near misses.

The expected perforations matrix described above is not the desired matrix; it merely shows the expected perforations sustained by a horizontal, square target of unit size (one cell) for every position of the target. To obtain the expected perforations matrix for the actual target plate, we superimpose on the actual target plate a grid having the same cell size and cell orientation as the quasi-infinite grid. The center cell of this actual target plate is defined by equations 4-7:

$$\begin{aligned} MIDX &= (NX \div 2) + 1 \\ MIDY &= (NY \div 2) + 1 \end{aligned} \qquad \text{Equations 4-7}$$

where

NX is the number of cells of the actual target plate along the range axis.

NY is the number of cells of the actual target plate along the deflection axis.

$\oplus$ is an operation defined as divide and then truncate the fractional part.

Defining the location of the actual target plate on the quasi-infinite plate as the location of its central cell on the quasi-infinite plate, we position the central cell of the actual target plate successively over each cell of the quasi-infinite plate and, for each such position, we sum the expected perforations of every cell covered by the actual target plate. The result is the expected perforations matrix for the actual target plate. This matrix is now multiplied by a scale factor, if desired, to account for any slight mismatch in surface areas that may exist between the actual target plate and the plate being modeled.

At this point, the expected perforations matrix is stored in the expected perforations matrix file and various types of information may be output, as will be discussed in paragraph 4.2.

4.1.2 Non-Horizontal Single Plates. The plates considered here are those obtained from a congruent, but horizontal, plate by a downward rotation from the horizontal about only one of the horizontal plate's four edges. Figure 4-2 shows two such plates rotated about the deflection axis. Figure 4-3 shows two such plates rotated about the range axis. The following sign conventions and constraints hold:

(1) $0° \leq |\alpha| < 180°$

(2) $0° \leq |\beta| < 180°$

(3) α is positive (negative) when a vector drawn outward and normal to the plate has a component in the direction of increasing (decreasing) range.

(4) β is positive (negative) when a vector drawn outward and normal to the plate has a conponent in the direction of increasing (decreasing) deflection.

(5) One or the other of α and β must be zero.

In this program, as in field artillery usage, deflection left is increasing and deflection right is decreasing. The treatment of such plates is similar to the treatment of the horizontal plate, but there is an increase in complexity.

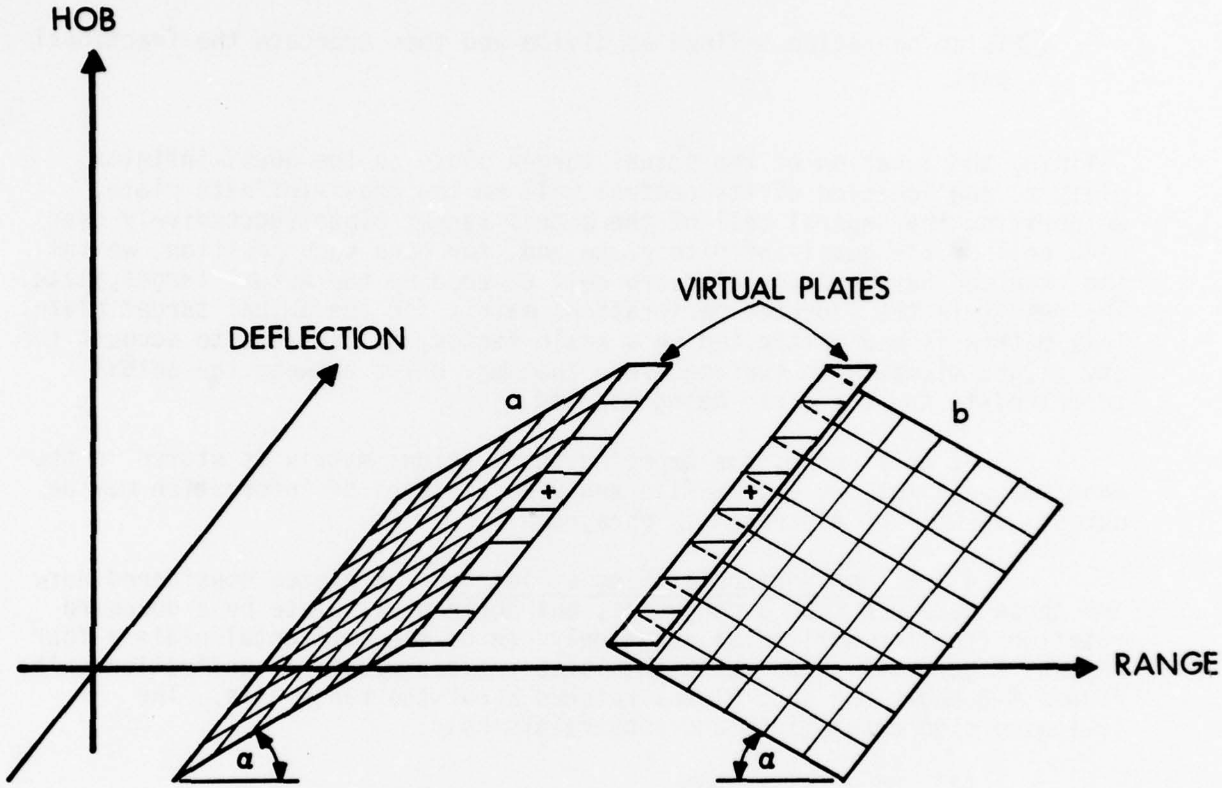


Figure 4-2. (U) Single Plates With a Rotation from the Horizontal About the Deflection Axis.

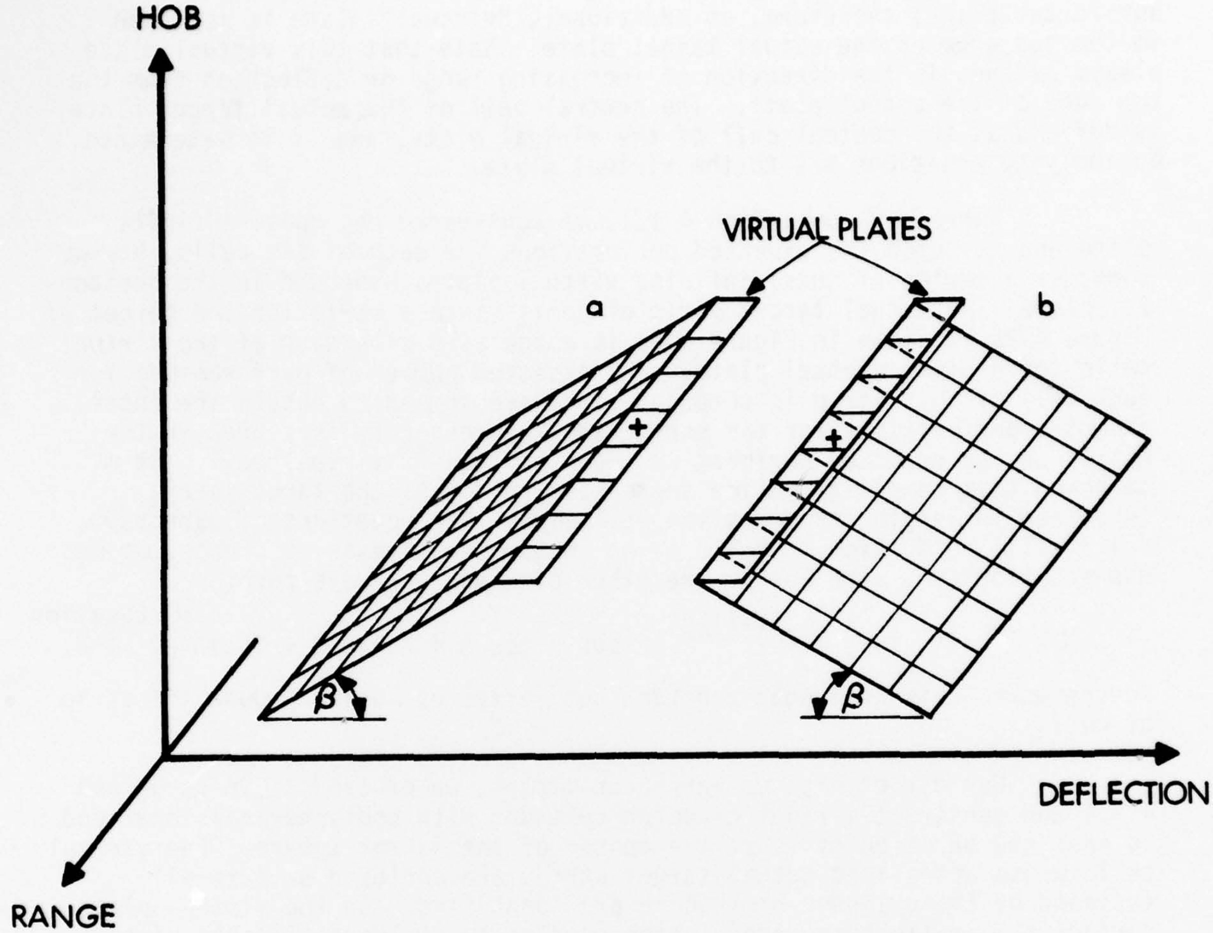


Figure 4.3 (U) Single Plates With a Rotation from the Horizontal About the Range Axis.

The target plate grid no longer meshes with the grid of the horizontal plane; therefore, an additional, "virtual" plate is appended to the top edge of the actual target plate. Note that this virtual plate always extends in the direction of increasing range or deflection from the top edge of the actual plate. The central cell of the actual target plate is defined as the central cell of the virtual plate, and it is determined by applying equations 4-7 to the virtual plate.

Whereas in paragraph 4.1.1, we considered one quasi-infinite plate and computed the expected perforations for each of its cells, now we consider a series of quasi-infinite virtual plates embedded in the horizontal plane. An actual target strip of cells (such a strip for the target of Figure 4-2a is shown in Figure 4-4) is associated with each of the virtual cells for a given virtual plate. The expected number of perforations for each cell of this strip is computed, they are summed to obtain the total expected perforations for the strip, and this quantity is stored in the matrix as the expected perforations for the $(i,j)^{th}$ virtual cell. It must be noted that computation time increases rapidly as the target strip increases in length for any given cell size. The equations of paragraph 4.1.1 still hold, except that ζ is no longer the same as Ψ. Hence, we must use equation 4-8, with one or the other of α and β always zero.

$$\cos \zeta = (X^2 + Y^2 + Z^2)^{-(1/2)}(Z \cos \alpha \cos \beta + X \sin \alpha + Y \sin \beta) \qquad \text{Equation 4-8}$$

Furthermore, Z is no longer constant but varies as we sweep down the strip of cells.

For direct hits or very near misses, we proceed as in paragraph 4.1.1 and construct a right circular cylinder with hemispherical lower end so that the burst point is at the center of the parent sphere. The virtual cells whose associated actual target strips are enclosed or partially enclosed by the cylinder-hemisphere are identified. On the virtual plate surface the smallest rectangle which completely encloses all such virtual cells is generated. The virtual cells enclosed by this rectangle are identified, and their associated actual target strips are the ones considered to sustain direct hits or very near misses.

As in paragraph 4.1.1, a sum over the virtual cells is performed to obtain the expected perforations matrix for the actual target plate. Of course, this sum is now 1-dimensional rather than 2-dimensional.

4.1.3 Non-Horizontal Plate Pairs. The concept of a non-horizontal plate pair arises from an examination of the two plates shown in Figure 4-2, or in Figure 4-3, and from the realization that if either of these plates is present in a lightly armored structure, it may well be accompanied by the other (e.g., the two sides of an APC). For this case, the two plates are considered as one plate pair and the symmetry of the situation is utilized to expedite the calculation. Except for the top spacer plate, nothing new is introduced. The utility of the concept is two-fold; it is faster on the computer than doing the two plates separately, and it expedites (for the machine) and facilitates (for the user) some of the subsequent computations of program section II.

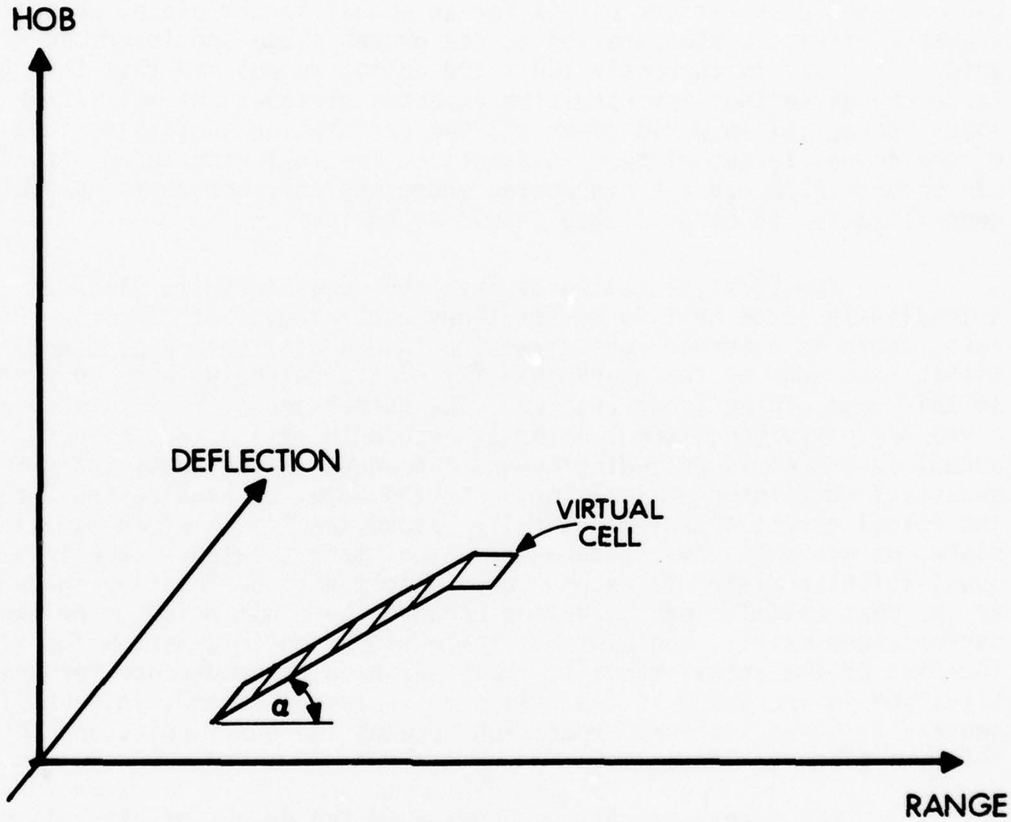


Figure 4.4. (U) A Target Strip and its Virtual Cell for the Target Plate of Figure 4.2a.

Figure 4-5 shows the two plates of Figure 4-2 combined to form a plate pair. The horizontal plate is not allowed to sustain perforations and serves merely as a spacer. The central cell of the plate pair is defined as the central cell of the top spacer and is computed from equations 4-7 applied to the top spacer. There is no longer a need for the virtual plate of paragraph 4.1.2 unless the top edges of the two non-horizontal plates are adjacent. In this case, the geometry can be visualized by letting the virtual plates of Figure 4-2 (or Figure 4-3) overlap each other completely. One of the angles, α and β, is always zero, the other is always positive and varies between 0° and 180°.

4.1.4 Problems Caused by Finite Grid Size. In order to obtain the expected perforations matrix for an actual target plate, we considered a quasi-infinite plate parallel to the ground plane and inscribed with a grid. The grid is currently 100 x 100 cells; we assumed that this grid was large enough so that the resulting expected perforations matrix for the actual target plate would cover all the perforation sustaining locations of the actual target plate. In practice, the following three situations can occur. Although the discussion addresses only horizontal plates, the generalization to other plates should be obvious.

The first situation is that the quasi-infinite plate is sufficiently large that no perforations occur beyond its limits. Furthermore, there is a border (whose required width will become apparent) just within each edge of the quasi-infinite plate, which sustains no perforations. In this case, there is no problem. The summations over the quasi-infinite plate are performed without errors (henceforth called edge effects) due to actual target cells extending beyond the edges of the quasi-infinite plate and still sustaining perforations. Furthermore, by considering locations of the actual target's center which lie beyond the limits of the quasi-infinite plate, we see that, even though the actual target extends back into the quasi-infinite plate, it extends only into the no perforation border. This is the most satisfactory situation because the final result, the expected perforations matrix, contains all the perforation information for every location of the actual target. The likelihood of occurrence for this situation is increased as the cell size is increased and, in general, decreased as the vertical separation between the burst point and the quasi-infinite plate is increased.

The second situation occurs when the quasi-infinite plate is sufficiently large that no perforations occur beyond its limits, but the no perforation border of the preceding situation either is absent or is not wide enough. An error called matrix truncation occurs (there are locations not represented by the expected perforations matrix for which the actual target will sustain perforations), but there are still no edge effects. This problem, if significant, can be eliminated by solving the same actual target again with a different projectile location with respect to the grid and manually piecing together the several expected perforations matrices, or by depending upon symmetry.

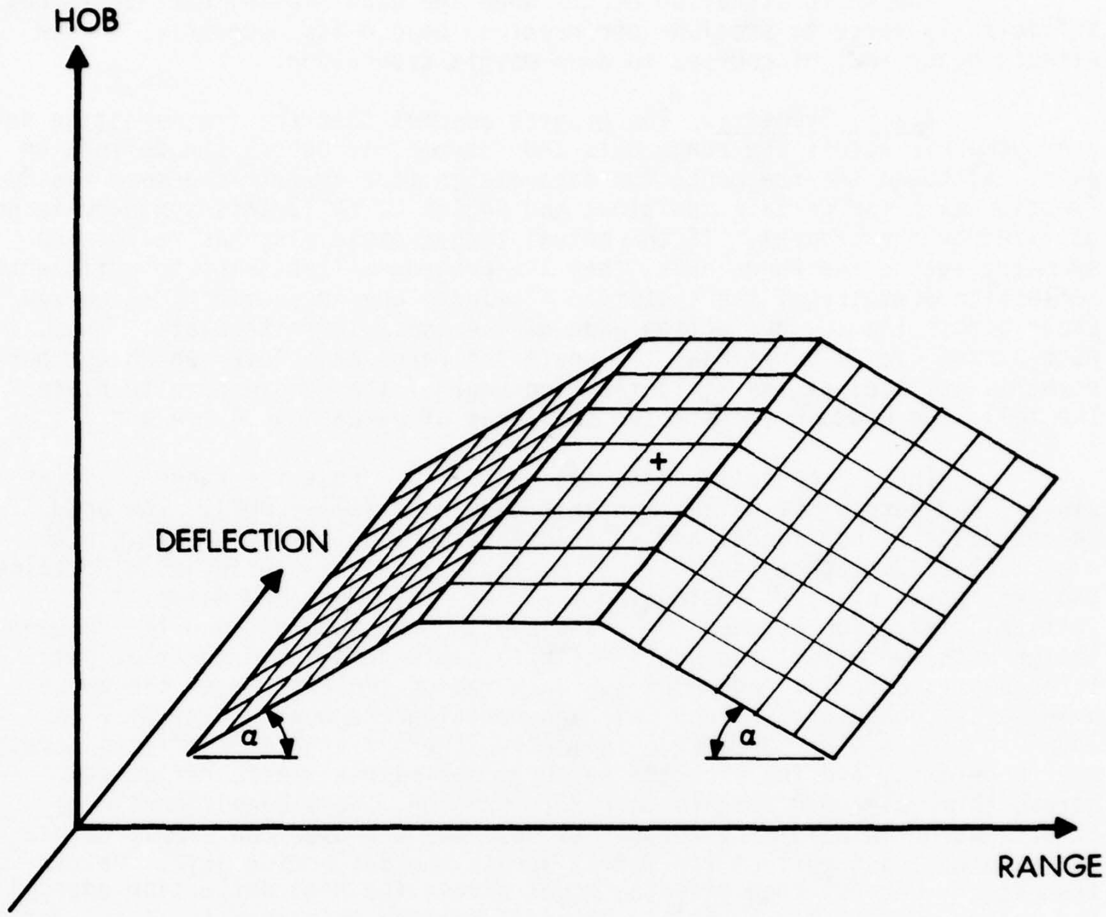


Figure 4.5. (U) A Plate Pair.

The third situation occurs when the quasi-infinite plate is not sufficiently large to preclude perforations beyond its boundaries. Edge effects occur and, of course, so does matrix truncation.

4.1.5 Symmetry. The program assumes that the fragmentation data are symmetric across the range axis and asymmetric across the deflection axis. Although the fragmentation data are in fact symmetric across the deflection axis for certain munitions and angles of fall, this symmetry is not utilized by the program. If the actual target plate also has reflection symmetry across the range axis, them the program will attempt to utilize the reflection symmetry of the system to eliminate any edge effects which may occur across the low deflection edge of the quasi-infinite plate. Thus, in such a case, it is beneficial to locate the range axis (over which the burst point is located) at the low deflection edge of the quasi-infinite plate. The following example illustrates the ideas of paragraphs 4.1.4 and 4.1.5.

The fragmentation data are symmetric across the range axis (as always) and across the deflection axis (angle of fall = 90°). The actual target plate is horizontal and so enjoys reflection symmetry across both axes. It is 36 inches long (along the range axis) and 20 inches wide (along the deflection axis). Considering the plate thickness and material, vertical separation between the plate and the burst point, and the fragmentation data, we expect the quasi-infinite plate to sustain perforations throughout a circular region of 400 inch radius centered under the burst point. We choose a four-inch cell and position the burst point at Z as shown in Figure 4-6. Because of symmetry, there are no edge effects across the range axis, and the expected perforations matrix can be reflected across this axis to eliminate matrix truncation. Very significant edge effects occur in columns 1 through 4; however, the user can simply delete these columns and reflect the matrix across the deflection axis. Matrix truncation, but not edge effects, occur across the high deflection edge of the matrix, but the error should be small because this area is at the far edge of the region in which perforations occur. Both matrix truncation and edge effects occur across the high range edge of the matrix but, again, the error should be small.

4.2 Output.

a. The input data are printed out.

b. The non-zero portion of the expected perforations matrix is written on the expected perforations file with a unique record number, which is the ordinal for that record. These records are sequentially and consecutively numbered for all cases within a batch (paragraph 4.3). Not more than 2500 records can be written on this file. If the entire matrix is zero, then a small zero matrix is written. If desired, the non-zero portion of the matrix can be printed. If the entire matrix is zero, its printing is suppressed. When printing occurs, each element is printed in an I6 format after being multiplied by a power of 10 (input data) and truncated.

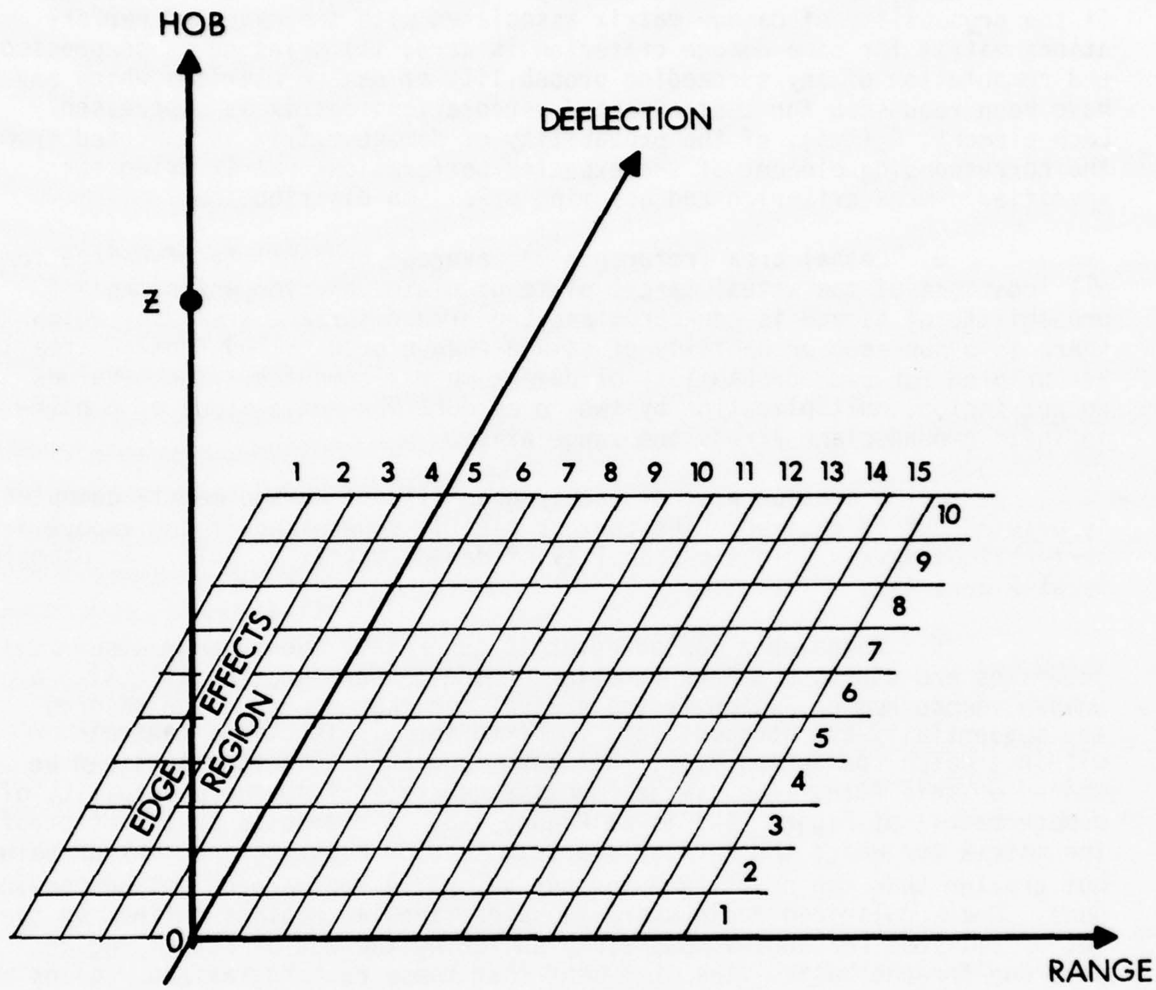


Figure 4.6. (U) A Situation Displaying Edge Effects.

c. The non-zero portion of the probability of damage matrix for each criterion is printed out if desired. If the expected perforations matrix is zero, none of its probability of damage matrices are computed. If the probability of damage matrix associated with the expected perforations matrix for some damage criterion is zero, its printing is suppressed and computation of any succeeding probability of damage matrices which may have been requested for that expected perforations matrix is suppressed. Each element, $P_d(i,j)$, of the probability of damage matrix is computed from the corresponding element of the expected perforations matrix using the specified damage criterion and assuming a Poisson distribution.

d. Lethal area (reference 3), average probability of damage for all locations of the actual target plate or plate pair for which the probability of damage is non-zero, and the ground surface area over which there is a non-zero probability of damage (henceforth called "damage area"), are printed for each probability of damage matrix computed. These values <u>do not</u> include multiplication by two to account for reflections of a half-infinite ground plane across the range axis.

e. A contour plot of each probability of damage matrix computed is printed out if desired. The contour plot is suppressed if the expected perforations matrix or the probability of damage matrix is null. An example is at Figure 4-7.

f. Whenever a contour plot is generated, the program also generates and stores a damage function on the damage function file with a unique record number, which is the ordinal for that record. The records are sequentially and consecutively numbered for all functions generated within a batch (paragraph 4.3). Not more than 5000 such functions can be stored on this file. The damage function generated from the probability of damage matrix of Figure 4-7 is at Figure 4-8. A region is that portion of the matrix for which the $P_d(i,j)$ are less than or equal to one contour value but greater than the next lower contour value. A region need not be continuous. Rectangularized regions are those rectangular regions defined by the set of smallest rectangles completely enclosing the set of actual regions on a one-for-one basis. The statement that these rectangularized regions are unnormalized means that lethal area and damage area are not conserved in their generation; hence, before they can be used, these damage functions must be normalized. The various regions in one damage function are written on the file from outermost to innermost. The maximum number of such regions is 11, corresponding to the regions within contour lines for P_d = 1., .9, .8, ..., .1, and .0005.

4.3 Input Data Deck

This deck inputs a "batch", or "logical grouping", of cases. All cases within a batch have the same cell size, ground plane grid size, projectile location with respect to the grid, and fragmentation data. For the purposes of this paragraph, a "target" is one plate or one plate pair. A "case", then, is one target at one plate thickness and one projectile HOB. Each target is solved for every specified thickness at every specified HOB.

```
PROBABILITY OF SUSTAINING AT LEAST  1 PERFORATIONS

HOB= 4.69 METERS     THICKNESS=1.00 INCHES     CELL= 8.00 INCHES     NX= 16 CELLS     NY=  9 CELLS     NZ=  0 CELLS     T=  78.0 INCHES

SCALE FACTOR APPLIED TO INPUT PLATE = 1./1.000

NXX=100 CELLS     NYY=100 CELLS     PROJECTILE LOCATION WAS (  55.5,     .5)

THE EXPECTED PERFORATIONS MATRIX FOR THIS CASE IS ON THE EXPECTED PERFORATIONS FILE (PMAT348    ) AS RECORD NUMBER     1

LETHAL AREA= 18.588 SQUARE METERS     AVERAGE PROBABILITY OF DAMAGE = .647 OVER    28.738 SQUARE METERS



                          OUTPUT OF THE PROBABILITY OF DAMAGE MATRIX HAS BEEN SUPPRESSED


THE PROBABILITY CONTOURS FOR THIS TARGET, HOB, AND DAMAGE CRITERION FOLLOW. THE NUMBERS ARE PD*10. (TRUNCATED), AND E=.0005, *=1.0

      4444444444555555555566666666667
      0123456789012345678901234567890

24    0CEEEEEEEEEEEEEEEEEE00000000000   24
23    E                  E          0   23
22    E                  E          C   22
21    E 1               1E          0   21
20    E12               21        E 0   20
19    123         4    432  1       E   19
18    124     5        543  2   1   E   18
17    1345   6          64  3 2  1  E   17
16    2456  7    8     876 54 3 2 1 E   16
15    3567  8     9    98 76 54 321 E   15
14    367 E  9          9 6 7654321 E   14
13    468 9               9 8765431 E   13
12    4789                 9 875431 E   12
11    589                   9876431 E   11
10    689       *      *     986532 E   10
 9    79    *           *    9875321E    9
 8    79   *             *    9763 2E    8
 7    89  *               *   9764 21    7
 6    89 *                 *  9764 31    6
 5    89*                  *  986 431    5
 4    9 *                  *  9865 42    4
 3    9 *                  *  987 542    3
 2    9 *                  *  987 542    2
 1    99*********************99875552    1

      4444444444555555555566666666667
      0123456789012345678901234567890
```

FIGURE 4-7. SAMPLE OUTPUT OF PROGRAM SECTION I

Unless otherwise stated, each data field is 10 columns wide and each numeric datum is in a real format with an explicit decimal point. Data fields not specifically addressed are ignored and may be used for any desired purpose, such as user comments within the data deck.

4.3.1 The Data Deck. The data deck for program section I is at Figure 4-9.

4.3.2 Definition of Each Input Variable

Card 1:

Columns 1-10 - the words "BASIC DATA".

FILES - the unique whole number identifying the first perforation matrix to be generated in this batch of cases. The perforation matrix file cannot contain more than 2500 records (matrices). FILES can be any whole number from 1 thru 2500. FILES is the first value of a consecutive series of numeric indices for the matrices to be generated and filed in the current batch of cases. FILES itself, however, need not be consecutive or even sequential with respect to the matrices stored in the perforation file by all preceeding batches. If FILES is not greater than the index of the highest indexed matrix placed in the file by all preceeding batches, some or all of the earlier matrices may be overlaid by matrices of the current batch. There is a default value for FILES. If FILES is blank or 0. its value is computed to be one greater than the index of the highest indexed matrix already in the file. If the file is empty, the default for FILES is 1.

CELL1 - the edge length, in inches, of the square cell which will be used to grid both the target and the ground plane.

NXX,NYY - the numbers of cells in the ground plane grid (and hence in the expected perforations matrix) along the range and deflection axes, respectively; $1. \leq NXX \leq 100.$; $1. \leq NYY \leq 100.$; in practice, these variables would be less than 100. only to save machine time.

ICOL,IROW - related to the range and deflection coordinates of the burst point in terms of ground plane grid columns and rows, respectively; grid columns increase with range and grid rows increase with deflection left; (ICOL,IROW) = (50.,0.) would yield burst point ground plane grid coordinates = (50.5,.5), corresponding to a location between the 50th and 51st grid columns and between the zeroeth (non-existent) and first grid rows; in practice, the usefulness of these variables is to shift the ground plane grid location with respect to the (fixed) burst point; a burst point outside the ground plane grid is allowed, with the exception stated in paragraph 5.3.

THESE DATA ARE ON THE DAMAGE FUNCTION FILE (DFCN348) AS RECORD NUMBER 1

AVERAGE PD WITHIN EACH REGION	LETHAL AREA IN METERS**2		DAMAGE AREA IN METERS**2	
	PER REGION	CUMULATIVE	PER REGION	CUMULATIVE
1.000	7.143	7.143	7.143	7.143
.971	5.731	12.874	5.905	13.048
.848	1.365	14.239	1.610	14.658
.749	.866	15.105	1.156	15.814
.651	.806	15.912	1.239	17.053
.551	.774	16.685	1.404	18.457
.449	.593	17.278	1.321	19.778
.354	.467	17.746	1.321	21.099
.249	.411	18.157	1.652	22.751
.146	.259	18.416	1.775	24.527
.041	.172	18.588	4.212	28.738

AVERAGE PD WITHIN EACH REGION	RECTANGULARIZED, UNNORMALIZED RANGE DATA				RECTANGULARIZED, UNNORMALIZED DEFLECTION DATA			
	EXTENT OF REGION		LOCATION OF REGION CENTER WRT PROJECTILE		EXTENT OF REGION		LOCATION OF REGION CENTER WRT PROJECTILE	
	METERS	CELLS	METERS	CELLS	METERS	CELLS	METERS	CELLS
1.000	4.267	21.0	-.711	-3.5	2.032	10.0	1.016	5.0
.971	5.080	25.0	-.711	-3.5	3.048	15.0	1.524	7.5
.848	5.283	26.0	-.610	-3.0	3.251	16.0	1.626	8.0
.749	5.486	27.0	-.508	-2.5	3.251	16.0	1.626	8.0
.651	5.486	27.0	-.508	-2.5	2.845	14.0	2.032	10.0
.551	6.096	30.0	-.203	-1.0	3.658	18.0	1.829	9.0
.449	6.096	30.0	-.203	-1.0	3.658	18.0	2.032	10.0
.354	6.096	30.0	-.203	-1.0	3.048	15.0	2.337	11.5
.249	6.299	31.0	-.102	-.5	4.064	20.0	2.032	10.0
.146	6.299	31.0	-.102	-.5	3.454	17.0	2.540	12.5
.041	6.299	31.0	-.102	-.5	3.454	17.0	3.150	15.5

FIGURE 4-6. DAMAGE FUNCTION PRODUCED FROM PROBABILITY OF DAMAGE MATRICES CORRESPONDING TO FIGURE 4-7

```
COLUMN           1111111111222222222233333333334444444444555555555566666666667777777777
NUMBER  1234567890123456789012345678901234567890123456789012345678901234567890123456789O
 CARD
NUMBER
   1    BASIC DATAFILES     CELL1     NXX       NYY       ICOL      IROW
   2    NHOB      HHHOB(1)  HHHOB(2)  HHHOB(3)  ...
               *** UP TO 3 MORE CARDS, AS NEEDED, FOR UP TO 25 VALUES OF HHHOB ***
   3    NTHCK     TTHCK(1)  TTHCK(2)  TTHCK(3)  ...
               *** UP TO 3 MORE CARDS, AS NEEDED, FOR UP TO 25 VALUES OF TTHCK ***
   4    NTRGT
   5    NXTGT(1)  NYTGT(1)  NZTGT(1)  NSIDE(1)  ALPHA(1)  BETA(1)   TT(1)     SC(1)SH(1)
   5    NXTGT(2)  NYTGT(2)  NZTGT(2)  NSIDE(2)  ALPHA(2)  BETA(2)   TT(2)     SC(2)SH(2)
                  *** UP TO 23 MORE CARDS, AS NEEDED, FOR UP TO 25 TARGETS ***
   6    NKRIT     KRIT(1)   KRIT(2)   KRIT(3)   ...
   7    THOR DATA
   8    METAL     CV        ALFAV     BETAV     GAMAV     LAMDAV    KFCTRV
   9              CM        ALFAM     BETAM     GAMAM     LAMDAM    KFCTRM
  10              VTERM     BLAST
  11    FRAGMENTATION DATA  SHAPE     AA
  12    NSPRA     NSP(1)    NSP(2)    NSP(3)    ...
              *** UP TO 5 MORE CARDS, AS NEEDED, FOR UP TO 46 VALUES OF NSP ***
  13    K         Q(1,K)    M(1,K)    Q(2,K)    M(2,K)    Q(3,K)    M(3,K)    Q(4,K)
  13    M(4,K)    Q(5,K)    M(5,K)    ...
                        *** THERE MAY BE UP TO 11 CARDS IN SET K TO
                            INPUT UP TO 40 VALUES OF Q(I,K), M(I,K) ***
             *** THERE MAY BE UP TO 46 SETS OF CARDS 14 TO INPUT NSFRA SPRAYS ***
  14    K         SPRBS(K)  SPRBS(K+1)VSTMN(K)  VSTMX(K)  NUM(K)    SGST(K)
                  *** THERE MAY BE UP TO 46 CARDS 13 TO INPUT NSPRA SPRAYS ***
  15    CONTOUR   NSTAPE    NO PROBABILITY MATRIX
```

FIGURE 4-9. DATA DECK FOR PROGRAM SECTION I

Card 2:

NHOB - the number of HOB's to be considered in this batch of cases; $1. \leq NHOB \leq 25.$

HHHOB(I)- the I^{th} HOB to be considered in this batch of cases; in meters.

As many as three additional cards may occur here to input up to 25 values of HHHOB; each additional card starts with a value of HHHOB in the first data field (columns 1-10).

Card 3:

NTHCK - the number of armor plate thicknesses to be considered in this batch of cases; each thickness is considered for each HOB; $1. \leq NTHCK \leq 25.$

TTHCK(I) - the I^{th} thickness to be considered in this batch of cases; in inches.

As many as three additional cards may occur here to input up to 25 values of TTHCK; each additional card starts with a value of TTHCK in the first data field (columns 1-10).

Card 4:

NTRGT - the number of targets to be considered in this batch of cases; $1. \leq NTRGT \leq 25.$; each target is considered for each thickness, thus, the number of cases in this batch is NTRGT*NTHCK*NHOB; an overall constraint is $1. \leq NTRGT*NTHCK*NHOB \leq 2500.$

Card 5:

There are NTRGT cards 5; each provides information for one target in this batch of targets; all targets will be solved for the first thickness at the first HOB, then for the second thickness at the first HOB, and so on until all thicknesses have been exhausted, then for the first thickness at the second HOB, and so on until all HOB's have been exhausted.

NXTGT(I) - the number of cells along the range axis in the I^{th} plate or in the spacer of the I^{th} plate pair.

NYTGT(I) - the number of cells along the deflection axis in the I^{th} plate or in the spacer of the I^{th} plate pair.

NZTGT(I) - the number of cells along the slope in the I^{th} plate or plate pair; zero or blank for horizontal plates.

NSIDE(I) - zero or blank for horizontal plates; 1. for single non-horizontal plates; 2. for plate pairs.

ALPHA(I) - 0. $\leq$ |ALPHA(I)| < 180.; the angle in degrees from the horizontal down to the outer surface of the I^{th} plate or plate pair; positive for plate pairs and plates whose outward drawn normal vector has a component in the direction of increasing range; negative otherwise; zero for plates rotated about the range axis and for horizontal plates.

BETA(I) - 0. $\leq$ |BETA(I)| < 180.; the angle in degrees from the horizontal down to the outer surface of the I^{th} plate or plate pair; positive for plate pairs and plates whose outward drawn normal vector has a component in the direction of increasing deflection (deflection left); negative otherwise; zero for plates rotated about the deflection axis and for horizontal plates.

TT(I) - the height of the top surface or top edge of the I^{th} plate or plate pair above the ground plane; in inches.

SC(I) - 5 column field; actually SCALE(I); a scale factor used to adjust for minor mismatch between actual and model plate surface areas; has no effect on direct hit contributions; equal to the surface area of the plate being input to the program divided by the surface area of the plate being modeled; default value is 1.

SH(I) - 5 column field; actually SHOW(I); non-negative whole number; the power to which 10. will be raised and then used to multiply each element of the expected perforations matrix prior to output of that matrix; if SH(I) = 0. or blank, the I^{th} matrix is not output; the output field for each element of the matrix is I6, including spaces between elements.

Card 6:

NKRIT - the number of damage criteria which will be considered for each case in this batch; 1. $\leq$ NKRIT $\leq$ 7.

KRIT(I) - the I^{th} damage criterion; KRIT(I) > 0; the input vector must be monotone increasing.

Card 7:

Columns 1-10 - the words THOR DATA; identifies the following two cards as providing Thor constants; from and in the same units as Thor reports 47, 70 (references 1,2); if not present, cards 8 and 9 also must be not present and Thor constants from the immediately preceding batch of cases will be used.

Card 8:

METAL - 10 alphanumeric characters; the user's own name for the material to which the Thor constants apply.

CV, ALFAV, BETAV, GAMAV, LAMDAV, KFCTRV - Thor constants for the residual velocity equations with no particular fragment shape assumed; KFCTRV is blank or 0. if the armor is not an aluminum for which K-factors are available, as in Thor report 70 (reference 2).

Card 9:

CM, ALFAM, BETAM, GAMAM, LAMDAM, KFCTRM - Thor constants for the residual mass equations with no particular fragment shape assumed; KFCTRM is blank or 0. if the armor is not an aluminum for which K-factors are available, as in Thor report 70 (reference 2).

Card 10:

W - the projectile angle of fall in degrees.

VTERM - The projectile terminal velocity in feet per second.

BLAST - radius in inches of the blast effects sphere, within which the probability of damage is 1.

Card 11:

Columns 1-10 - the word, FRAGMENTAT; identifies subsequent cards as providing the fragmentation distribution; if not present, cards 12 through 17 also must not be present and the fragmentation data from the immediately preceding batch of cases, or from the immediately preceding execution of program Section IV, will be used.

SHAPE - the fragment shape factor; in grains/cubic inch.

AA - a constant in the velocity decay equation (see equation 4-5).

Card 12:

NSPRA - the number of fragment sprays in the fragmentation data; $3. \leq NSPRA \leq 46.$

NSP(I) - the number of mass values in the I^{th} spray; this vector may be continued on as many as 5 additional cards, starting in column 1 of each, for NSPRA elements in NSP; $1. \leq NSP(I) \leq 40.$, all I.

Card 13:

There may be as many as 46 sets of as many as 11 cards each to input as many as 40 Q(J,K), M(J,K) pairs for each of the NSPRA fragment sprays; K runs from 1 to NSPRA; J runs from 1 to NSP(K).

K - the spray number.

Q(J,K) - the fraction of all fragments in the K^{th} spray residing in the J^{th} mass interval.

M(J,K) - the mass assigned to fragments residing in the J^{th} mass interval of the K^{th} fragment spray; in grains; for any given K, the M(J,K) must be monotone increasing.

Card 14:

There may be as many as 46 cards #14 to define NSPRA fragment sprays.

K - the spray number.

SPRBS(K), SPRBS(K+1) - the leading and trailing edges of the K^{th} fragment spray; in degrees; the leading (trailing) edge of the first (last) spray must be 0.° (180.°); the sprays must be continuous from 0° through 180°.

VSTMN(K) - the median fragment velocity at the midpoint of the K^{th} fragment spray; in feet per second; the midpoints of the first and last sprays are defined as 0.° and 180.°, respectively.

VSTMX(K) - the maximum fragment velocity at the midpoint of the K^{th} fragment spray; in feet per second; the midpoints of the first and last sprays are defined as 0.° and 180.°, respectively.

NUM(K) - the number of fragments in the K^{th} fragment spray; if 0. or blank, will be computed from SGST(K).

SGST(K) - the density of fragments (number per steradian) in the K^{th} fragment spray; if 0. or blank, will be computed from NUM(K).

Card 15:

An optional card which may use either of two formats. The first format is:

Columns 1-10 - the word CONTOUR; causes a contour map of the probability of of damage matrix to be output for each case and damage criterion of the current batch; also causes generation of a damage function, storage thereof on the damage function file, and printed output thereof.

NSTAPE - the unique whole number identifying the first damage function to be generated in this batch of cases. The damage function file cannot contain more than 5000 records (functions). NSTAPE can be any whole number from 1. thru 5000. NSTAPE is the first value of a consecutive series of numeric indices for the functions to be generated and filed in the current batch of cases. NSTAPE itself,

however, need not be consecutive or even sequential with respect to the functions stored in the damage function file by all preceeding batches. If NSTAPE is not greater than the index of the highest indexed function placed in the file by all preceeding batches, some or all of the earlier functions may be overlaid by functions of the current batch. There is a default value for NSTAPE. If NSTAPE is blank or 0., its value is computed to be one greater than the index of the highest indexed function already in the file. If the file is empty, the default for NSTAPE is 1.

Columns 21-30 - the words NO PROBABI; causes suppression of the printing of all the probability of damage matrices for this batch.

The alternate format is:

Columns 1-10 - the words NO PROBABI, as above.

Columns 31-40 - the word CONTOUR, as above.

Columns 41-50 - the variable NSTAPE, as above.

5. PROGRAM SECTION II - VULNERABILITY OF ARMORED STRUCTURE

5.1 The Program.

The function of program Section II is to assemble the expected perforations matrices (generated by Section I and retrievable from the perforation matrix file) for as many as seven plates or plate pairs to obtain the expected perforations matrix, probability of damage matrices, and damage functions for one complete target. All the matrices which are combined for any one case must be for the same cell size and projectile HOB, angle of fall, and terminal velocity. There can be up to 25 cases in one batch. The manner in which the expected perforations matrices associated with several plates are combined to form the expected perforations matrix for a complete target is best described through the use of an example.

5.1.1 Assembling a Complete Target. Consider the target shown in Figure 5-1, which shows a side view with elevation, a side view, and a front view. Plates seen on edge have tick marks to indicate cell boundaries. The extent of plates seen on edge is indicated. The scale and coordinate system are common to all three views. Data for each plate or plate pair are shown in Table 5-1.

Plate A is embedded in the spacer BB which can sustain no perforations. Between the edges of plate A and the top edges of sides B is a region which could correspond to heavily armored parts of an actual structure. The shaded region is the virtual plate associated with plate C. Reflection symmetry exists across the range axis.

Figures 5-2, 5-3, and 5-4 provide examples of additional plates which could be part of the complete structure, but which are not considered in the current example. Partial data are given in Table 5-1. These plates are shown to indicate how reasonably complex structures can be built up and to show various possible values of ALPHA, BETA, AND TT. The points Q, R, and S of Figures 5-2, 5-3, and 5-4 correspond to those same points in Figure 5-1.

Consider plate A with central cell D and plate pair B with central cell E. We define plate A as the reference plate for the structure; its central cell will become the central cell for the entire structure.

Using the ground plane grid of Figure 4-6, element (I,J) of the appropriate expected perforations matrix is the expected number of perforations sustained by plate A when its central cell (D) is positioned over cell (I,J) of the ground plane. Similarly, element (I+1,J) of another expected perforations matrix is the expected number of perforations sustained by plate pair B when its central cell (E) is positioned over cell (I+1,J) of the ground plane. Thus, for all rows J and all columns I:

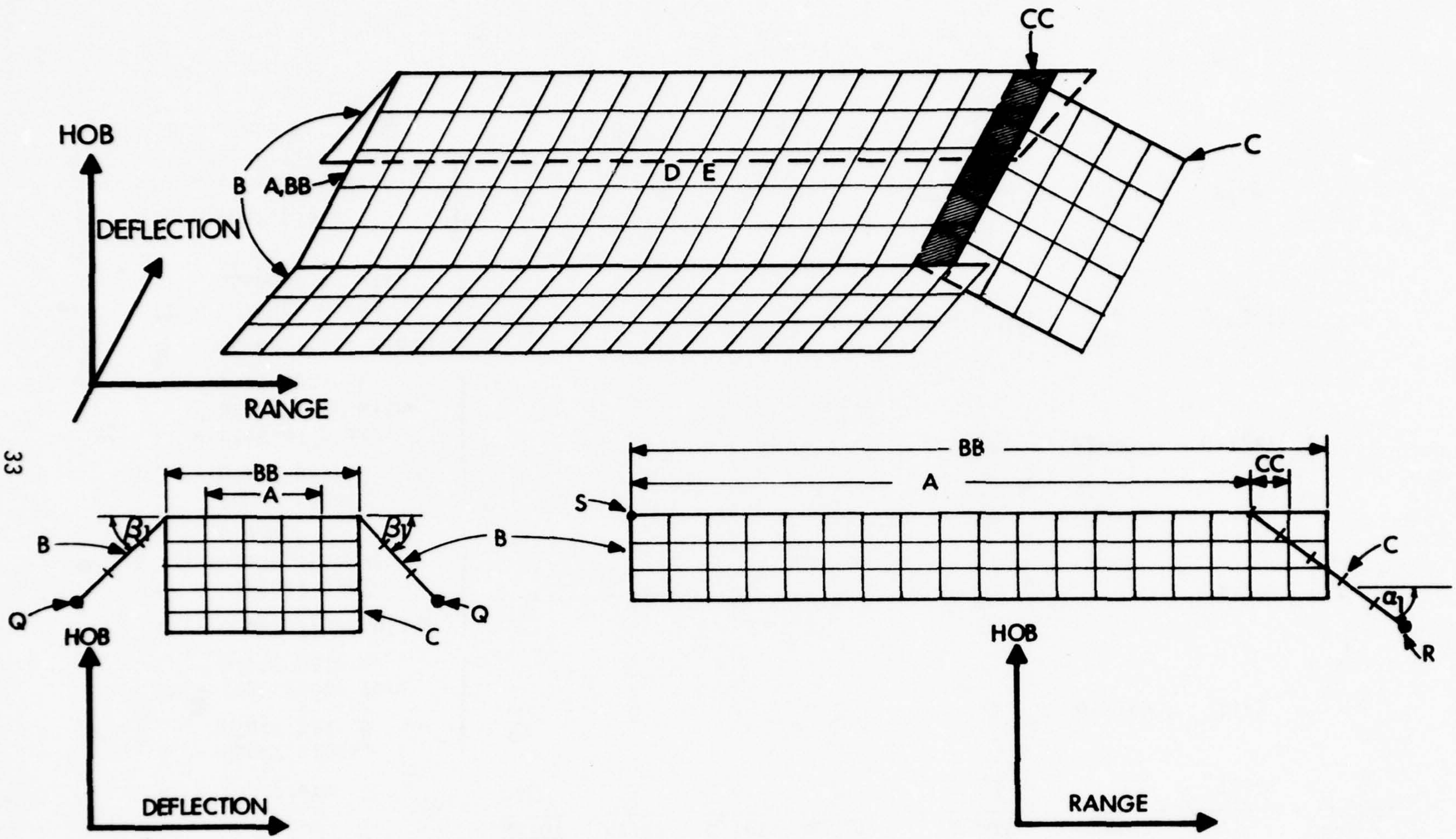


Figure 5.1 (U) Model of a Complete Target.

TABLE 5-1. DATA FOR PLATES OF FIGURES 5-1 THROUGH 5-4

	NXTGT	NYTBT	NZTGT	NSIDE	ALPHA	BETA	TT	CENTRAL CELL
A - top plate	16	3	0	0	0	0	T1>0	D
B - upper sides, plate pair B; BB - top spacer for plate pair B	18	5	3	2	0	$0<\beta 1<90°$	T1>0	E
C - upper front plate; CC - virtual plate associated with plate C	0	5	5	1	$0<\alpha 1<90°$	0	T1>0	F
L - lower sides, plate pair; LL - virtual plate associated with plate pair L				2	0	$\beta 2=90°$	0<T2<T1	
M - lower front plate; MM - virtual plate associated with plate M	0			1	$90°<\alpha 2<180°$	0	0<T3<T1	
N - rear plate; NN - virtual plate associated with plate N	0			1	$-90°>\alpha 3>-180°$	0	T1>0	

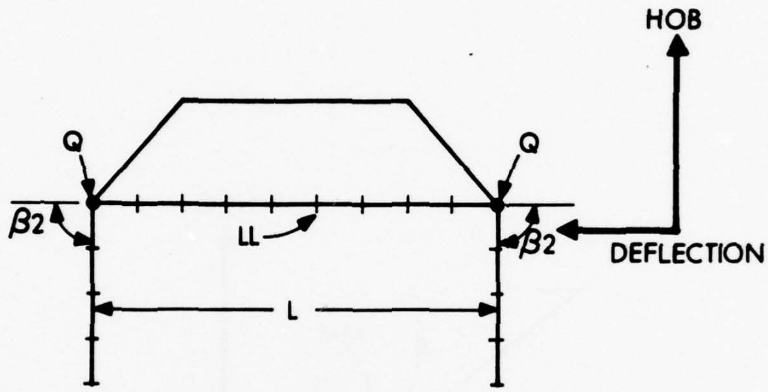


Figure 5.2 (U) Lower Sides Which Could be Added to the Model of Figure 5.1.

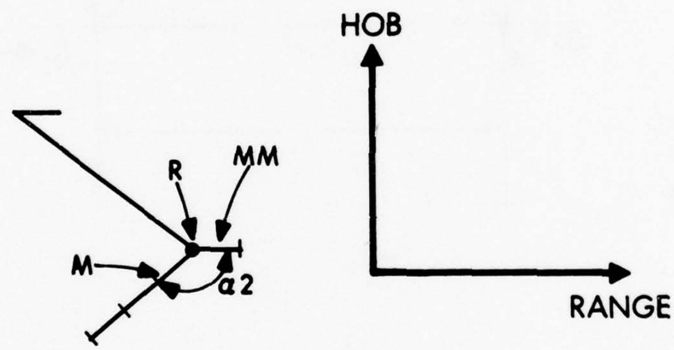


Figure 5.3 (U) Lower Front Plate Which Could be Added to the Model of Figure 5.1.

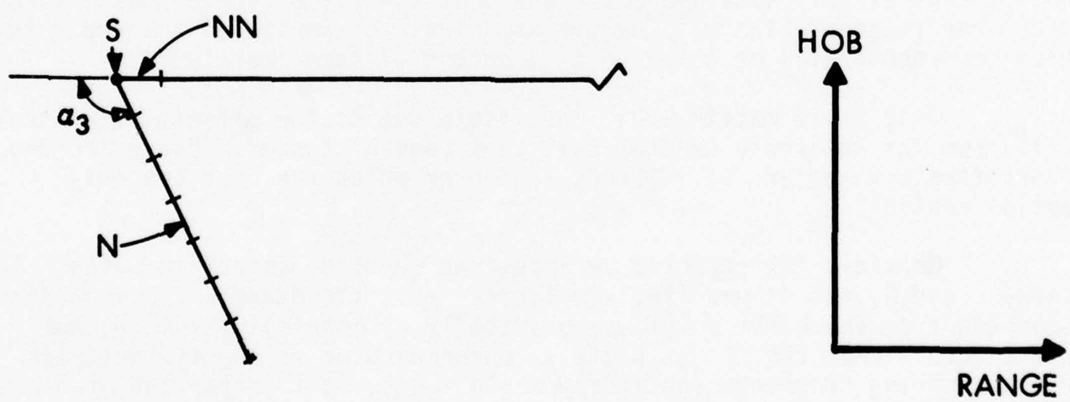


Figure 5.4. (U) Rear Plate Which Could be Added to the Model of Figure 5.1.

$$EPM_{A,B}(I,J) = EPM_A(I,J) + EPM_B(I+1,J) \qquad \text{Equation 5-1}$$

where $EPM_{A,B}(I,J)$ means element (I,J) of the expected perforations matrix for the structure made up of plate A and plate pair B.
Similarly,

$$EPM_{A,B,C}(I,J) = EPM_{A,B}(I,J) + EPM_C(I+8,J) \qquad \text{Equation 5-2}$$

or

$$EPM_{A,B,C}(I,J) = EPM_A(I,J) + EPM_B(I+1,J) + EPM_C(I+8,J) \quad \text{Equation 5-3}$$

Thus, the expected perforations matrix for a complete structure is obtained by adding, with appropriate offsets, the expected perforations matrices for individual components.

It would be most efficient to first compute the offsets in range and deflection and then add all the matrices together, element by element, at once, as indicated by Equation 5-3. This, however, was beyond the storage capability of the computer for which this program was originally developed (BRLESC I/II of the USABRL); therefore, Equations 5-1 and 5-2 are used. The first two components are combined to obtain an intermediate structure which is then treated as a component with which the next component is combined, and so on. Therefore, the first component (the one identified by logical record number FILE(1)) is always used as the reference component and its central cell always becomes the central cell of the complete structure. For reasons which will become apparent, the choice of the order in which components will be combined is a matter of some importance.

Only those matrix addition offsets due to the offsets of central cells from the reference central cell have been discussed. There are two other offsets which are of interest, and over which the user has only partial control.

Consider the expected perforations matrices associated with plates A and C, and assume (for simplicity) that the dynamic fragmentation projected from the burst point is spherically symmetric. Since maximum damage occurs when the struck plate is perpendicular to the trajectories of the striking fragments (obliquity angle = 0°), it is clear that the region of maximum damage for plate C will be in lower numbered columns than it will be for plate A, for a wide range of HOB's. Hence, the user well may have positioned the projectile over different grid points when computing EPM_A and EPM_C in order to minimize matrix truncation (paragraph 4.1.1) for each of these plates. The program automatically compensates for any such offsets in projectile locations.

The following describes the second type of offset. Assume NXX = NYY = 100 for each expected perforations matrix. The various offsets

applied when combining these component matrices may result in an expected perforations matrix for the complete structure which extends beyond one or more sides of the NXX x NYY cell grid. Each time a component matrix is added into the result of previous additions (or into the reference matrix, if the addition is the first one to be performed), the program will attempt to minimize matrix truncation by rigidly translating the entire plates/projectile system within the grid. However, the program always gives preference to the prior matrices over the later matrices; that is, regardless of the magnitudes of the matrix elements being lost, and regardless of the number of such elements, the program will never shift any part of a prior matrix or matrix sum out of the grid in order to accommodate the matrix currently being added in. Therefore, consideration must be given to the order in which the component matrices are presented to the program (i.e.: the order of the NFILE elements of the set {FILE(I)}; see paragraph 5.4) and the dominant matrices should be presented first.

5.1.2 Symmetry and an Example. As in the case for Section I of the program, Section II can make use of symmetry across the range axis, and the user can utilize symmetry across the deflection axis. Consider the structure of Figure 5-1 as a target oriented in the direction of increasing range. Clearly, reflection symmetry exists across the range axis for each target component and also for the complete target. Therefore, the burst point for each component should be located between ground plane grid rows 0 (non-existent) and 1 so that each component's expected perforations matrix will contain maximum information. As the target is assembled, there will be no need to shift the burst point from this zero deflection location (shifts in range may occur) and the final matrix also will contain maximum information. A similar discussion applies to the same vehicle oriented in the direction of decreasing range.

Rotate the structure of Figure 5-1 clockwise by 90° and consider it as a target oriented across the range axis in the direction of decreasing deflection (Figure 5-5). (Note that the virtual plate belonging to plate C has changed its orientation with respect to plate C, and now lies on top of one end of plate A.) Since the fragmentation is symmetric across the range axis, it is clear that the solution over the entire ground plane differs from that for the same target oriented in the direction of increasing deflection only by a reflection across the range axis. Thus, we need only one of the two orientations for the target broadside to the range axis. We cannot, however, proceed (as in the preceding paragraph) to solve over the half-infinite ground plane, and depend on reflection symmetry of the target itself to obtain the complete solution, because target reflection symmetry does not exist. The problem can be overcome in either of two ways. The first is to position the burst point somewhere over the center of the ground plane grid. This is completely satisfactory so long as the user can stand the (extreme) matrix truncation which may occur. The other method will be discussed in paragraph 5.3. At this point, we present an example to illustrate the solution over the half-infinite plane of the target facing in the direction of decreasing deflection.

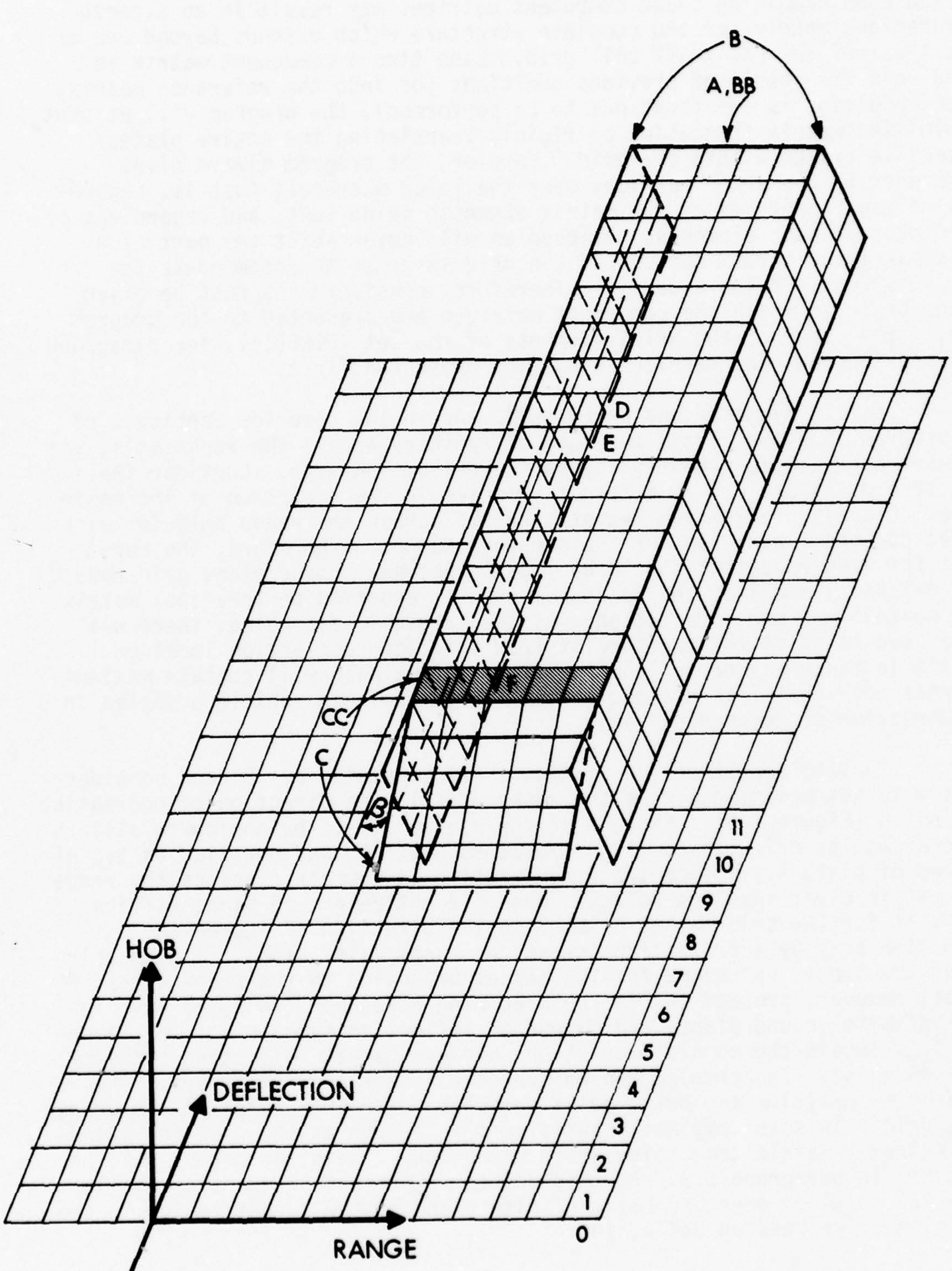


Figure 5.5 (U) The Target Model of Figure 5.1 Oriented in the Direction of Decreasing Deflection.

We have stored in a file the expected perforations matrices for each target component (plate A, plate pair B, and plate C of Figure 5-5). When these matrices were computed, we chose the projectile burst locations with some care; for components A and B, it was between grid rows 0 (non-existent) and 1, and for component C, it was between grid rows 9 and 10. Plate A is considered the reference plate and its central cell is the central cell for the entire structure. As stated previously, the perforation matrices for the several components are additive so long as the proper offsets are maintained. To determine which element of a component's perforation matrix should be added to an element of the reference plate's or reference structure's perforation matrix, we use Equation 5-4 and a similar equation for matrix columns.

(component matrix row to be added in)

= (structure matrix row under consideration)

+ (row offset of component central cell with respect to structure central cell)

+ (row offset of burst point in component matrix with respect to burst point in structure matrix)

Equation 5-4

For plate pair B, the last two terms of Equation 5-4 are -1 and 0, respectively. Furthermore, there are no column offsets. Therefore,

$$EPM_{A,B}(I,J) = EPM_A(I,J) + EPM_B(I,J-1) \qquad \text{Equation 5-5}$$

We note that when cell D is in grid row 1, there is no information available for plate pair B and hence we would have an edge effect error, except for the fact that plate pair B enjoys reflection symmetry across the range axis.

A general statement is that, if matrix B has central cell row offset, K, and burst point row offset, L, then

$$EPM_{A,B}(I,J) = EPM_A(I,J) + EPM_B(I,J+K+L) \qquad \text{Equation 5-6}$$

Furthermore, if component B enjoys reflection symmetry across the range axis and if $K < 0$ and $K + L < 0$, then, when $J \leq |K + L|$, Equation 5-6 is replaced by Equation 5-7.

$$EPM_{A,B}(I,J) = EPM_A(I,J) + EPM_B(I,1-(J+K)+L) \qquad \text{Equation 5-7}$$

Returning to the specific case at hand, we have Equation 5-5 which holds except when cell D is in grid row 1. When cell D is in grid row 1, we use Equation 5-8, obtained from Equation 5-7.

$$EPM_{A,B}(I,J) = EPM_A(I,J) + EPM_B(I,2-J) \qquad \text{Equation 5-8}$$

Thus far, we have obtained the expected perforations matrix for the structure of Figure 5-5, not including plate C. The structure's central cell is cell D. The range axis is still located between grid rows 0 (non-existent) and 1. The structure does not enjoy reflection symmetry across the range axis (because plate pair B sticks out in front of plate A, but not in back of it), but this is of no importance because the structure is not added into anything; rather, components are added into the structure. To add plate C into the structure, we use equation 5-9, obtained from equation 5-6 with K = -9 and L = 9

$$EPM_{A,B,C}(I,J) = EPM_{A,B}(I,J) + EPM_{C}(I,J) \qquad \text{Equation 5-9}$$

It should now be clear that the entire purpose for locating the range axis for plate C between grid rows 9 and 10 was to avoid edge effects due to plate C when the central cell of the structure is near the burst point. Equation 5-7 could not have been used because plate C does not enjoy reflection symmetry across the range axis.

5.2 Output.

Program status at this point is very similar to its status after computations for a target component have been completed by Section I. The output is the same as in paragraph 4.2 with the following exceptions:

a. The expected perforations matrix for the complete structure is not stored on the perforation matrix file.

b. There is no suppression of probability of damage matrices or of the contour plots and damage functions due to null matrices.

5.3 An Additional Capability for Targets Displaying Asymmetry Across the Range Axis.

Consider the target of Figure 5-5 and allow it to assume both orientations perpendicular to the range axis in both half-infinite parts of the ground plane. Figure 5-6 illustrates the four possibilities. For each case, the target is symmetric across the deflection axis and asymmetric across the range axis. The fragmentation is symmetric across the range axis and asymmetric across the deflection axis. The expected perforations matrices for A and D are reflections of each other, and similarly for B and C. Furthermore, the matrices representing the complete solutions (A,C and B,D) are reflections of each other. Hence, the solutions for A and B contain all the information required for a solution of either orientation over the entire ground plane. The program has the capability of performing the symmetry operations required to piece together the matrices A and for B, thereby obtaining the complete solution for both orientations. The output, including lethal area, average probability of damage, and damage area, will pertain to the complete ground plane. To use this capability, we must have $0 \leq IROW < NXX$ (see paragraph 4.3.2, card 1 for IROW) for both of the complete structure matrices, A and B. Furthermore,

both values of IROW should be small because the portions of the two matrices A and B in rows numbered less than or equal to the corresponding IROW value are lost as the two matrices are pieced together. For a similar reason, ICOL should be approximately the same for both matrices.

Note that this treatment depends upon the reflection symmetry of the structure across the deflection axis. If the structure suffers asymmetry across that axis, a very similar treatment applies, but the result has a different interpretation. In this event, structures B, D are not actual structures with orientation opposite to A,C. Rather, they are hypothetical structures which are reflections of A,C across the range axis. Furthermore, after the matrices for A and B are pieced together at the range axis, the resulting matrix A,B represents only A and C. If orientation B and D is desired, then the solution for a second set of conditions (structure B, reflected structure A) must be obtained.

5.4 Input Data Deck.

This deck inputs a batch of cases. A case is one target (usually consisting of more than one component), projectile, burst point combination. A batch may contain as many as 25 cases and, although there is usually some relationship among the cases within a batch, there need not be. If the capability of paragraph 5.3 is used, the cases must come in pairs, and there can be 25 such pairs. In this situation, the comments previously made pertaining to a case pertain to a case pair.

Unless otherwise stated, each data field is 10 columns wide and each numeric datum is in a real format with an explicit decimal point. Data fields not specifically addressed are ignored and may be used for any desired purpose.

5.4.1 The Data Deck. The data deck for program Section II is at Figure 5-7.

5.4.2 Definition of Each Input Variable.

Card 1:

Columns 1-10 - the word ASSEMBLE.

KSYM - KSYM = 2. causes the union of 2 expected perforations matrices at their range axes to represent, over the entire ground plane, a complete target; otherwise ignored.

NPNTS - has two somewhat different interpretations; if KSYM $\neq$ 2., NPNTS is the number of cases in this batch, a case being one complete target at one set of projectile conditions; if KSYM = 2., NPNTS is the number of case pairs in the batch; in either case, 1. $\leq$ NPNTS $\leq$ 25.

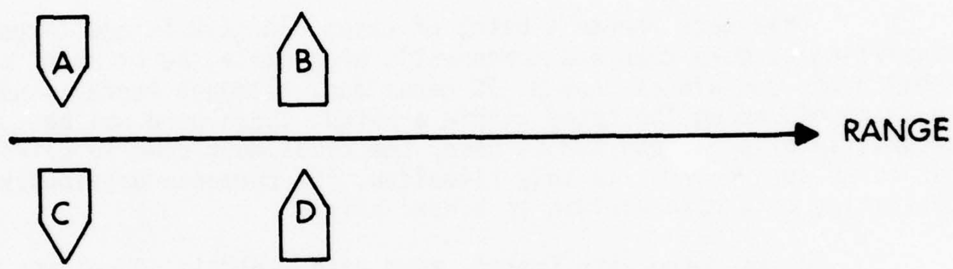


Figure 5-6. (U) Possible Target Orientations.

```
COLUMN              111111111122222222223333333333444444444455555555556666666666777777777778
NUMBER     12345678901234567890123456789012345678901234567890123456789012345678901234567890
 CARD
NUMBER
   1       ASSEMBLE  KSYM       NPNTS      CONTOUR   NSTAPE    NO PROBABILITY MATRIX
   2       EXPECTED  PERFORATIONS ALL=
   3       SHOW(1)    SHOW(2)   SHOW(3)    ...
                *** UP TO 3 MORE CARDS, AS NEEDED, FOR UP TO 25 VALUES OF SHOW ***
   4       NKRIT       KRIT(1)   KRIT(2)   KRIT(3)   ...
   5       NFILE      FILE(1)   FILE(2)    FILE(3)   ...
   6       SHIFT X =            SHFTX(2)   SHFTX(3)  ...
   7       SHIFT Y =            SHFTY(2)   SHFTY(3)  ...
                               *** UP TO 24 MORE SETS OF THE ABOVE 3
                                   CARDS, AS NEEDED, FOR UP TO 25 CASES ***
```

FIGURE 5-7. DATA DECK FOR PROGRAM SECTION II

Columns 21-30 - the word CONTOUR; causes a contour map of the probability of damage matrix to be output for each case and damage criterion of the current batch; also causes generation of a damage function, storage thereof on the damage function file, and printed output thereof; otherwise, ignored.

NSTAPE - the unique whole number identifying the first damage function to be generated in this batch of cases. The damage function file cannot contain more than 5000 records (functions). NSTAPE can be any whole number from 1. thru 5000. NSTAPE is the first value of a consecutive series of numeric indices for the functions to be generated and filed in the current batch of cases. NSTAPE itself, however, need not be consecutive or even sequential with respect to the functions stored in the damage function file by all preceding batches. If NSTAPE is not greater than the index of the highest indexed function placed in the file by all preceding batches, some or all of the earlier functions may be overlaid by functions of the current batch. There is a default value for NSTAPE. If NSTAPE is blank or 0., its value is computed to be one greater than the index of the highest indexed function already in the file. If the file is empty, the default for NSTAPE is 1.

Columns 51-60 - the words NO PROBABI; causes suppression of the printing of the probability of damage matrices for this batch; otherwise, ignored.

Card 2:

This is an optional card.

Columns 1-10 - the words EXPECTED P; causes output of the expected perforations matrices for individual cases of this batch as indicated below:

Columns 23-30 - in columns 23-26, ALL= and in columns 27-30, a whole number > 0., which number is the power to which 10. will be raised and then used to multiply each element of the expected perforations matrix; the fractional part of each element will be truncated and the matrix will be output in I6 format.

Card 3:

This card is present only if card 2 is present and columns 23-26 thereof are not as described above. There must be NPNTS values of SHOW(I), where SHOW(I) is defined in paragraph 4.3.b., card 5; any Ith matrix for which SHOW(I) = 0. will not be output.

Card 4:

NKRIT - the number of damage criteria which will be considered for each case in this batch; $1 \leq$ NKRIT ≤ 7.

KRIT(I) - the I^{th} damage criterion; KRIT(I) > 0.

Card 5:

Cards 5, 6, and 7 form a set which completely defines one case in this batch of cases. Either or both of cards 6 and 7 may be omitted for any card 5, in which case the values from the immediately preceding card 6 or 7 are used. If cards 6 or 7 or both are missing for the first case of the batch, the values are taken to be zero. If KSYM = 2., a special rule applies. There must be 2 sets of cards 5,6,7 to define a case; the first triplet defines the first half of the case pair and the second triplet, the second half. Both cards 5 must be present for the case, even if they are identical to each other. Cards 6, 7 may be present or absent with the same effect between the halves of a case and between cases as occurred between cases for KSYM ≠ 2.

NFILE - the number of logical records (expected perforations matrices for individual plates or plate pairs) required to obtain the expected perforations matrix for the complete target specified by this set of cards 5, 6, and 7; $1. \leq$ NFILE ≤ 7.

FILE(I) - the unique logical record number associated with the expected perforations matrix stored on the expected perforations file for the I^{th} plate (or plate pair) of the current target.

Card 6:

Columns 1-10 - the words SHIFT X =

SHFTX(I) - the number of cells by which the central cell of the I^{th} component plate (or plate pair) of this complete target is offset from the central cell of the first component of this complete target along the range axis; I must run from 2 to NPNTS.

Card 7:

Columns 1-10 - the words SHIFT Y =

SHFTY(I) - same as SHFTX(I), but along the deflection axis; recall that deflection left is positive and right, negative.

6. PROGRAM SECTION III - EFFECTIVENESS SUBMODEL

6.1 The DAMAGE FUNCTION.

The damage functions generated by program sections I and II are not suitable for use by this program section because they are not normalized, because their various regions may not be nested in a manner which can be handled by this program section, and because they may represent only one half of a damage function. They are, however, a very convenient point of departure for the construction of usable damage functions.

There are three types of geometry which are allowed for the regions of the damage functions. They are identified as fully nested, unnested, and partially nested, and are exemplified in Figures 6-1, 6-2, and 6-3, respectively. With respect to Figure 6.3, note that the damage function must be considered as having seven regions rather than five. The .9 and .1 regions each must be divided into two regions at the range axis so that they can be represented by rectangles. There are no restrictions concerning how the probability of damage may vary from one region to the next; however, there is a rule concerning geometrical arrangement of regions. The rule is that any rectangular contour which causes nesting to occur must fully enclose all preceding contours, and each subsequent region of that damage function must completely enclose all regions preceding it. Once the geometry of the final damage function has been determined, it must be normalized. That is, the lethal area and damage area for any region of the rectangularized, normalized damage function must be the same as the sums of the lethal and damage areas, respectively, over the elements of the probability of damage matrix which were combined to obtain that region. It should be clear that a probability of damage matrix or, more conveniently, an unnormalized damage function from program section I or II, can result in more than one damage function as used by section III. Furthermore, the choice of the damage function's final form is a matter of judgement.

To illustrate the generation of fully nested damage functions, we first obtain from the unnormalized damage function of Figure 4-8 what is commonly called a "cookie cutter" damage function. Such a damage function consists of one region with some constant probability of damage, and the entire exterior region has zero probability of damage. The total lethal and damage areas are 18.588 and 28.738 square meters, respectively, so the average probability of damage (abbreviated PD) is .65. The only quantities remaining to be defined are the deflection to range ratio of the rectangle and the location of the rectangle's center with respect to the burst location. We choose the deflection to range ratio to be the same as it is for the PD = .354 region (3.048/6.096 = .500) and we choose the range coordinate of the cookie cutter's center to be the same as for the PD = .971 rectangle (.711 meters closer to the cannon than is the burst point). Because the target plate was horizontal and rectangular, the probability of damage matrix (Figure 4-7 shows the contour map) corresponds to a target enjoying reflection symmetry across the range axis, and so the lethal area, damage area, and deflection to range ratio must be doubled and the deflection coordinate of the cookie cutter's center must be set to zero.

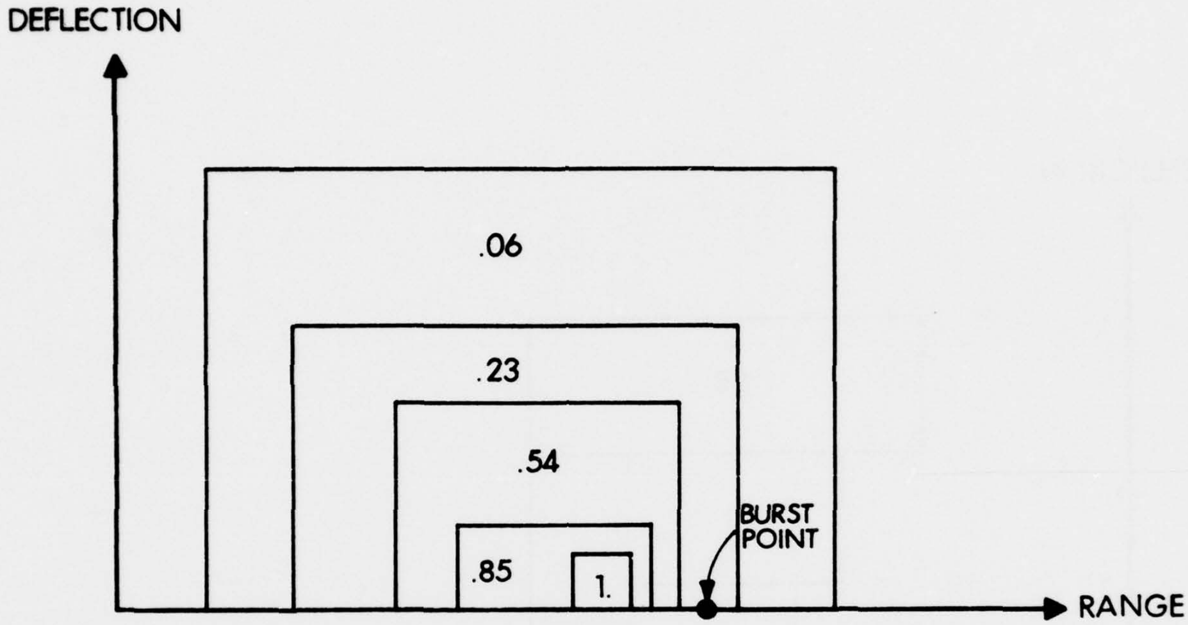


Figure 6-1. (U) One Half of a Fully Nested Normalized Damage Function Enjoying Reflection Symmetry Across the Range Axis.

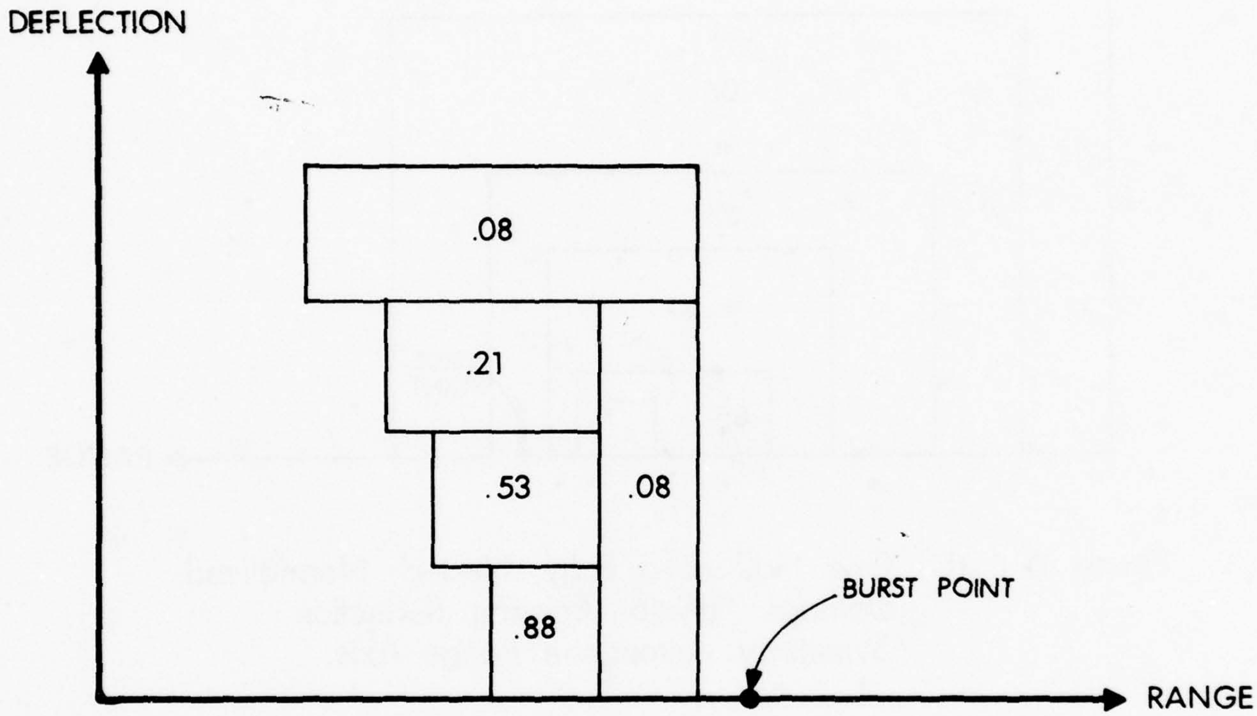


Figure 6-2. (U) One Half of an Unnested Normalized Damage Function Enjoying Reflection Symmetry Across the Range Axis.

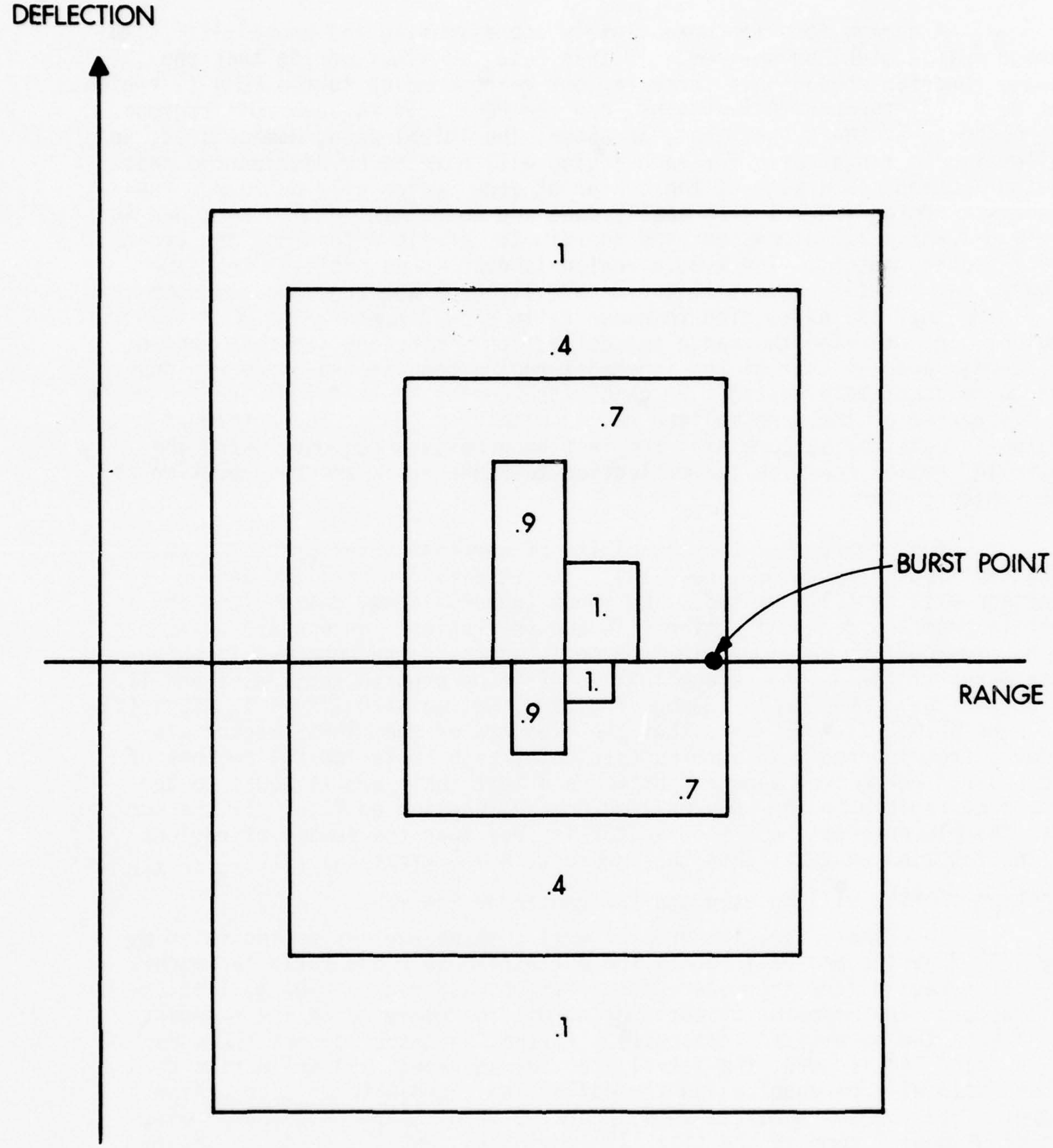


Figure 6-3. (U) A Complete Partially Nested Normalized Damage Function, Assymetric Across the Range Axis.

A damage function more closely approximating the probability of damage matrix might be desired. In this case, we might decide that the damage function should have three regions corresponding to the PD = 1. region, the PD = .971 through .449 regions, and the PD = .354 through .041 regions. We recognize at the outset that, as above, the lethal area, damage area, and deflection to range ratio for each region will have to be doubled and that the deflection coordinate of the center of each region will be zero. The innermost region (PD = 1.) is easily done and we obtain results as shown in Table 6-1, where all dimensions and coordinates are in meters and the areas are in square meters. The middle region is done in essentially the same manner, but now six regions of the unnormalized damage function are combined. We decide that the deflection to range ratio should be from the PD = .551 region. In obtaining the range and deflection dimensions, we must augment the damage area by that of the innermost region because the three regions are to be completely nested. We choose the center of the middle region to be at the center of the unnormalized region with PD = .848. The outermost region is obtained by combining the last unnormalized regions, using the PD = .146 region for both the deflection to range ratio and the location of the region center.

The program has the capability of performing the preceding computations (paragraph 6.4.2, card 14). The resulting normalized damage function will be fully nested. The input (unnormalized) damage function is read in from the damage function file and the regions thereon are assumed to be ordered from outermost to innermost, which is the sequence in which they were written on the damage function file by program sections I and II. To obtain the three region damage function, we use STEPS2(1) = 1., STEPS2(2) = 6., and STEPS2(3) = 4. Note that the elements of the STEPS2 vector are ordered from innermost to outermost (usually high PD to low PD) regions of the desired normalized damage function and that their sum is equal to the number of regions in the unnormalized damage function on file. If the sum over the elements of the STEPS2 vector is less than the number of regions in the function on file, then the innermost $N = \sum_J \text{STEPS2}(J)$ regions in the function on file will be used and the remainder ignored.

In either case, the program will combine regions as indicated by the STEPS2 vector and will obtain the deflection to range ratio for each resulting region from the outer boundaries of that region. Normalization will occur. The midpoint of each region will be computed as the midpoint of each of the normalized rectangles. If the "symmetry" option (paragraph 6.4.2, card 14) is used, the lethal area, damage area, and deflection to range ratio will be doubled and the deflection coordinate of each region's midpoint will be set equal to zero. Working from innermost to outermost, the program will compare the $(i + 1)^{th}$ region to the i^{th} region to insure that all regions are nested. If the $(i + 1)^{th}$ rectangle cannot enclose the i^{th} rectangle because of shape, the shape of the $(i + 1)^{th}$ rectangle will be altered just enough, maintaining normalization, to allow it to enclose the i^{th} rectangle. Also, if the $(i + 1)^{th}$ rectangle must be shifted to enclose the i^{th} rectangle, the $(i + 1)^{th}$ midpoint will be shifted just enough for nesting to occur.

TABLE 6-1. A FULLY NESTED DAMAGE FUNCTION

Region	Lethal Area	Damage Area	Cumulative Damage Area	Prob of Damage	Deflection to Range Ratio	Normalized Range Dimension	Normalized Deflection Dimension	Coordinates of Region Center With Respect to Projectile
Innermost	14.286	14.286	14.286	1.000	.952	3.874	3.688	(-.711,0.)
Middle	20.270	25.270	39.556	.802	1.200	5.741	6.890	(-.610,0.)
Outermost	2.620	17.920	57.476	.146	1.096	7.242	7.936	(-.102,0.)

For unnested damage functions, and for the unnested part of a partially nested damage function, the procedure is similar, but less complicated simply because we need not be concerned about nesting. However, the program has no capability to generate such damage functions.

In any event, the program reflects the complete damage function in its final form through the projectile burst point. This transforms the function from one describing the damage sustained by a target for every location of that target about the burst point into a function describing the damage sustained by a target due to a projectile burst at any point about that target, with constant HOB.

6.2 The Program.

This section of the program uses the methodology of reference 4, with some changes in the computer program to increase flexibility and delete unneeded portions, to compute the probability that each target in a group of several similar targets is damaged. This probability is stored internally as an element of a 3-dimensional matrix, henceforth called the S matrix. The three subscripts of these matrix elements identify the target element sustaining damage, the size of the volley which caused the damage, and the HOB of all projectiles in the volley.

Essentially, the S matrix is computed as follows. The specified number of projectiles is fired at the selected aim points; proper launch is assumed. Round-to-round and occasion-to-occasion error (assumed normal) and projectile functioning reliability (binomial) distributions are applied to the projectiles. HOB is constant. A distribution of burst points over the specified target array is obtained. The burst point distribution and the damage function representing the target element are combined and integrated over the ground plane. The result is one element of the S matrix. Results are obtained for each volley size, each target in the target array, and each HOB (each HOB has its own damage function) to obtain all elements of the S matrix.

Elements of the S matrix can be averaged over any specified HOB distribution (so long as the distribution includes only HOB's for which there are elements) to obtain a two-dimensional S' matrix. The two subscripts of these matrix elements identify the target element sustaining damage and the size of the volley causing the damage.

Optimization can be performed upon the S matrix for a specified volley size to determine which HOB causes the most damage. This HOB is then treated as an HOB distribution with standard deviation equal to zero to obtain an S' matrix. The optimization consists of summing the S matrix elements over the various targets for each HOB at the specified volley size. The HOB yielding the largest sum is defined as optimum for that volley size.

From the S' matrix are obtained, for each volley size specified, the probabilities that various numbers of targets in the target array sustain damage, the average probability (averaged over the target elements)

that some target element sustains damage, and the most and least damageable target elements with the probability that they sustain damage.

Consider the plane of elements in the S matrix corresponding to one target element. Each such plane is generated by a machine computation which can be time consuming. Under certain conditions, the number of such planes to be computed can be halved and the remainder obtained through an appeal to reflection symmetry. The conditions are that the damage functions for all specified HOB's, the aim point array, and the target array enjoy reflection symmetry across the range axis. Note that, apart from being common to both the aim points and the target elements, the Cartesian coordinate system used is arbitrary and its origin may be chosen in any convenient manner.

When these conditions are satisfied, the complete aim point array is input, but only one of the two symmetric halves of the target array is input, and the SYMMETRIC ARRAY option of input data cards 16 and 19 through 21 (Figure 6-4) is used. This allows the use of a target array containing twice as many target elements (50) as can be input (25), and results in some economy of machine time.

6.3 Output.

The input data are printed out. The S and S' matrices are printed out. Depending upon the number of targets in the target array, such of the following as apply are printed out for all specified volley sizes at the specified HOB distribution (the optimum HOB, if computed, being considered as an HOB distribution with zero standard deviation). These and the S' matrix are printed out only when requested by the appropriate cards 16 through 21 (see paragraph 6.4.2), whereas the S matrix is always printed out.

a. Probability that no targets are damaged.

b. Probability that exactly one target is damaged.

c. Probability that exactly two targets are damaged.

d. Probability that at least one target is damaged.

e. Probability that at least two targets are damaged.

f. Probability that at least three targets are damaged.

g. Probability that less than two targets are damaged.

h. Probability that less than three targets are damaged.

i. Average probability that a target is damaged.

j. Probability that the most vulnerable target is damaged.

k. Probability that the least vulnerable target is damaged.

6.4 Input Data Deck.

This deck inputs a batch of cases. Data cards 1 through 15 provide data used by all cases in the batch. These data include the number of cannon and their aim points, the number of targets in the target array and their locations, the number of rounds fired by each cannon for each of the volley sizes, ballistic data, functioning reliability, and damage functions for each of as many as 25 HOB's. Data cards 16 through 18 or cards 19 or 20 or 21 define a specific case within the batch. As many cases as desired may be run within one batch. A case now means a specific HOB distribution consisting of some or all of the HOB's for which damage functions were input.

Unless otherwise stated, each data field is 10 columns wide, and each numeric datum is in a real format with an explicit decimal point. Data fields not specifically addressed are ignored and may be used for any desired purpose.

6.4.1 The Data Deck. The data deck for program section III is at Figure 6-4.

6.4.2 Definition of Each Input Variable.

Card 1:

Columns 1-10 - the word ENCOUNTER

N - the number of cannon to fire; $1. \leq N \leq 100.$

NVEH - the number of targets in the target array to be fired upon; $1. \leq NVEH \leq 25.$; see card 16, columns 31-40.

NVS - the number of volley sizes to be solved; $1. \leq NVS \leq 25.$

NHTS - the number of HOB's, and hence, the number of damage functions to be solved; $1. \leq NHTS \leq 25.$

U1 - the number of increments for the range and deflection integrals over a quarter of the integration plane; usually, $U1 \geq 8.$

Cards 2 and 3:

A(I), B(I) - the range and deflection coordinates, respectively, of the aiming point for the I^{th} cannon; in meters; recall that deflection left is positive and right, negative.

There may be as many as 13 cards for the A's and the same number for the B's to allow the input of N values of A and B.

Card 4:

An optional card; has the words SAME CANNO in columns 1-10. If present, cards 2 and 3 must be absent. Causes the aiming points of the immediately preceding section III data deck to be used.

```
COLUMN            1111111111222222222233333333334444444444555555555566666666667777777777 8
NUMBER   12345678901234567890123456789012345678901234567890123456789012345678901234567890
 CARD
NUMBER
   1     ENCOUNTER N         NVEH      NVS       NHTS      U1
   2     A(1)      A(2)      A(3)      ...
                *** UP TO 12 MORE CARDS, AS NEEDED, FOR UP TO 100 VALUES OF A ***
   3     B(1)      B(2)      B(3)      ...
                *** UP TO 12 MORE CARDS, AS NEEDED, FOR UP TO 100 VALUES OF B ***
   4     SAME CANNON
   5     VOSR(1)   VOSR(2)   VOSR(3)   ...
                *** UP TO 3 MORE CARDS, AS NEEDED, FOR UP TO 25 VALUES OF VOSR ***
   6     VOSD(1)   VOSD(2)   VOSD(3)   ...
                *** UP TO 3 MORE CARDS, AS NEEDED, FOR UP TO 25 VALUES OF VOSD ***
   7     SAME TARGETS
   8     NN(1)     NN(2)     NN(3)     ...
                 *** UP TO 3 MORE CARDS, AS NEEDED, FOR UP TO 25 VALUES OF NN ***
   9     SAME VOLLEYS
  10     S1        S2        T1        T2        R
  11     SAME ERRORS
  12     HT(I)     NUMB(1)   NUMNN(I)
  13     XSSP(I,J) XBB3(I,J) XBB4(I,J) XROSE(I,J)XDDSE(I,J)
                       *** AS MANY AS 12 MORE CARDS, AS NEEDED, FOR UP TO
                           13 SETS OF THE ABOVE 5 VARIABLES, INDEXED ON J ***
              *** THE ABOVE 2 TO 14 CARDS FORM A SET DEFINING 1 DAMAGE FUNCTION ***
  14     **********SYMMETRY  FILE2ST(1)ST(2)ST(3)...
                         *** THE ABOVE CARD DEFINES A DAMAGE FUNCTION TO
                             BE EXTRACTED FROM THE DAMAGE FUNCTION FILE ***
             *** ADDITIONAL CARDS AND/OR CARD SETS, AS NEEDED, TO DEFINE A TOTAL
                 OF AS MANY AS 25 DAMAGE FUNCTIONS, EACH FOR A DIFFERENT HT(I) ***
  15     SAME DAMAGE FUNCTIONS
  16     HOB DISTRIBUTION    NHOB      SYMMETRIC ARRAY     OPTIMIZE  L
  17     HOB(1)    HOB(2)    HOB(3)    ...
                *** UP TO 3 MORE CARDS, AS NEEDED, FOR UP TO 25 VALUES OF HOB ***
  18     DF(1)     DF(2)     DF(3)     ...
                 *** UP TO 3 MORE CARDS, AS NEEDED, FOR UP TO 25 VALUES OF DF ***
                    *** AS MANY SETS OF THE ABOVE 3 TO 9 CARDS AS DESIRED ***
  19     ALL HOBS                      SYMMETRIC ARRAY     OPTIMIZE  L
  20     SAME HOB DISTRIBUTION         SYMMETRIC ARRAY     OPTIMIZE  L
  21     OPTIMIZE  L                   SYMMETRIC ARRAY
```

FIGURE 6-4. DATA DECK FOR PROGRAM SECTION III

Cards 5 and 6:

VOSR(I), VOSD(I) - the range and deflection coordinates, respectively, of the I^{th} target in the target array; in meters; recall that deflection left is positive and right, negative.

There may be as many as four cards for the VOSR's and the same number for the VOSD's to allow for the input of NVEH values of VOSR and VOSD. See card 16, columns 31-40.

Card 7:

An optional card; has the words SAME TARGE in columns 1-10. If present, cards 5 and 6 must be absent. Causes the target locations of the immediately preceding section III data deck to be used.

Card 8:

NN(I) - the number of rounds to be fired by each cannon in the I^{th} volley size to be solved.

There may be as many as four cards for the NN's to allow for the input of NVS volley sizes.

Card 9:

An optional card; has the words SAME VOLLE in columns 1-10; if present, card 8 must be absent. Causes the volley sizes of the immediately preceding section III data deck to be used.

Card 10:

S1, S2 - precision errors; the range and deflection standard deviations of the distribution of one cannon's burst points at one HOB on one occasion about their own mean; in meters.

T1, T2 - mean point of impact (MPI) errors; the range and deflection standard deviations of the distribution of one cannon's single occasion mean burst points at one HOB about their own mean; in meters.

R - the complete round functioning reliability, given proper launch.

Card 11:

An optional card. If present, card 10 must be absent; has the words SAME ERROR in columns 1-10. Causes S1, S2, T1, T2 and R from the immediately preceding section III data deck to be used.

Card 12:

Cards 12 and 13 define one damage function in final form. There must be NHTS sets of cards 12 and 13. These sets need not be in any particular order.

HT(I) - the HOB to which the I^{th} damage function applies; any two elements must differ from each other by at least .1% of their sum; in meters.

NUMB(I) - the number of regions in the I^{th} damage function; $1. \leq$ NUMB(I) ≤ 13.

NUMNN(I) - the number of unnested regions in the I^{th} damage function; $0. \leq$ NUMNN(I) $\leq$ NUMB(I).

Card 13:

XSSP(I,J) - the average probability of damage over the J^{th} region of the I^{th} damage function.

XBB3(I,J), XBB4(I,J) - the range and deflection dimensions, respectively, of the J^{th} region of the I^{th} damage function; in meters.

XROSE(I,J), XDOSE(I,J) - actually XROSET(I,J), XDOSET(I,J); the range and deflection offsets, respectively, of the center of the J^{th} region of the I^{th} damage function with respect to the target location; in meters.

There must be NUMB(I) cards 13 in the I^{th} set of cards 12 and 13. Their order must be from innermost to outermost damage function region. The order among themselves of those cards defining unnested regions is unimportant, but all such unnested cards must occur before any cards defining nested regions in the I^{th} set.

Card 14:

One card 14 replaces one set of cards 12 and 13. Any and any number of those card sets may be so replaced. Causes input of data from the damage function file, from which data the program will generate a fully nested damage function.

Columns 1-10 - any alphanumeric character string which cannot be interpreted as a number.

Columns 11-20 - if columns 11-20 contain the word SYMMETRY, the generated damage function will be assumed to be half of a damage function enjoying reflection symmetry across the range axis and it will be reflected across the range axis to produce a complete function. Lethal area and damage area will double. Otherwise, ignored.

FILE2 - 5 column whole number; the unique logical record number identifying the desired data on the damage function file.

ST(J) - 5 column whole number; actually STEPS2(J); one value for each region desired in the normalized fully nested damage function to be generated; each value indicates the adjacent regions of the unnormalized damage function on file which will be combined to arrive at the J^{th} region of the desired normalized damage function; the elements of this set must be ordered from innermost to outermost regions of the desired function; the sum of all the STEPS2(J) must not be greater than the number of regions in the function on file (not more than 11 regions); if the sum of all the STEPS2(J) is less than the number of regions in the function on file, then only the innermost $N = \sum_J$ STEPS2(J) regions of the damage function on file will be used; the first STEPS2(J) = 0. or blank terminates the vector. There is one exception to the preceding rules: if <u>all</u> the STEPS2(J) are 0. (or blank) then no regions on $\overline{\text{file}}$ will be combined: that is, the result is identical with the STEPS2 vector being all 1.'s and having as many elements as there are regions in the damage function on file.

Card 15:

An optional card. If present, all cards 12, 13, and 14 must be absent. The words SAME DAMAGE in columns 1-10 cause the entire set of normalized damage functions from the immediately preceding section III data deck to be used.

Card 16:

Cards 16, 17, 18 form a set defining one HOB distribution. There may be as many or as few such sets as desired, or none.

Columns 1-10 - the words HOB DISTRI.

NHOB - the number of HOB's in the distribution; $1. \leq \text{NHOB} \leq 25.$

Columns 31-40 - the word SYMMETRIC; causes the target array to be reflected across the range axis, thus producing a new target array enjoying reflection symmetry across the range axis and having twice as many target elements as the array which was input; can be used only when the aimpoints and the damage function are symmetric across the range axis; does not carry over from one case to the next or from one batch to the next; otherwise, ignored.

Columns 51-60 - the word OPTIMIZE; causes HOB optimization without prior use of an HOB distribution.

L - the number of rounds fired per cannon at which optimization will be performed; if L is not a member of the NN(I) vector, NN(1) will be used; ignored if columns 51-60 are ignored.

Cards 17 and 18:

HOB(I), DF(I) - the I^{th} HOB, and its weighting factor, respectively, for the current HOB distribution function; each HOB must be within .1% of an element of the HT(I) vector.

There may be as many as four cards 17 and four cards 18 to allow the input of NHOB values of HOB and corresponding values of DF.

Card 19:

Cards 19, 20, and 21 each perform a function equivalent to that of the set of cards 16, 17, and 18. As many or as few as desired may be used and they may be intermingled with each other and with the set of cards 16, 17, and 18, in any order desired (but not <u>inside</u> the set 16, 17, and 18), with the obvious restriction on card 20, that there must have been an immediately preceding case.

Columns 1-10 - the words, ALL HOBS; causes the current HOB distribution to be a uniform distribution consisting of all HOB's in the set {HT}.

Columns 31-80 - same as card 16.

Card 20:

Columns 1-10 - the words, SAME HOB D; causes the current HOB distribution to be the same as in the immediately preceding case; the immediately preceding case can be the last case of the immediately preceding batch.

Columns 31-80 - same as card 16.

Card 21:

Columns 1-10 - the word, OPTIMIZE; causes HOB optimization without prior use of an HOB distribution.

L - same as card 16.

Columns 31-50 - same as card 16.

7. PROGRAM SECTION IV - PERFORATIONS; POSSIBILITY AND SIGNIFICANCE

7.1 The Program.

This program section provides two functions not directly related to those of program sections I, II, and III. It predicts, for an armor plate and a fragment distribution, whether perforations can ever occur except for direct hits and very near misses. If perforations can occur, it computes the average probability of incapacitation of a soldier protected against the fragment distribution by the armor plate, assuming that the soldier will be hit by the distribution of residual fragments. Once again, the methodology is that of Thor Reports 47 and 70 (references 1 and 2). Air drag is neglected. For those fragments which perforate, the residual mass and velocity are computed. For each perforating fragment, the methodology of reference 5 is used to obtain the probability of incapacitation of a soldier, given that he is hit by the residual fragment. This methodology is augmented by the computation of a V_{50} using the equations of reference 8. Correspondence of the naked soldier and the soldier wearing a helmet and winter uniform in reference 5 to the soldier wearing a summer uniform and the soldier wearing a winter uniform in reference 8, respectively, is assumed. These $P_{I/H}$'s are averaged over the distribution of perforating fragments to obtain the average $P_{I/H}$ per fragment for the soldier protected by the armor plate.

7.2 Output.

The input data are printed out. Computed output is shown in Figure 7-1.

7.3 Input Data Deck.

This data deck is identified by its first card which contains the words SECTION IV in columns 1-10. After this card are input any number of "batches" of cases. One batch is terminated by a card with the word COMPUTE in columns 1-10. One case is one armor plate or plate assembly at one obliquity angle. For any given batch, the input data are divided into six data card subsets, which are the fragmentation data, terminal velocity, Thor constants, angles, armor, and incapacitation constants subsets. These subsets may be presented in any order.

All six subsets must be present for the first batch, except that, if the desired fragmentation data are currently in storage from program section I (see paragraph 8), then the fragmentation data subset need not be present. For subsequent batches, only those data card subsets whose input data are to be changed need be present.

Unless otherwise stated, each data field is 10 columns wide, and each numeric datum is in a real format with an explicit decimal point. Data fields not specifically addressed are ignored and may be used for any desired purpose.

7.3.1 The Data Deck. The data deck for program section IV is at Figure 7-2.

AL 7039   1.25 INCHES THICK   THETA = 0.00 DEGREES   PROJECTILE VELOCITY = 915.00 FEET PER SECOND

SPRAY	V IMPACT	M IMPACT	V RESIDUAL	M RESIDUAL	PI/H
6	.4592E+04 (MAX)	.6370E+03	.9798E+02	.4415E+03	0.0000
8	.4594E+04 (MAX)	.1649E+04	.1481E+04	.1212E+04	.9991
8	.4055E+04 (MED)	.1649E+04	.8879E+03	.1304E+04	.9930
9	.4745E+04 (MAX)	.1352E+04	.1400E+04	.9588E+03	.9974
9	.4207E+04 (MED)	.1352E+04	.8043E+03	.1039E+04	.9840
9	.4745E+04 (MAX)	.9690E+03	.9405E+03	.6723E+03	.9765
9	.4207E+04 (MED)	.9690E+03	.3375E+03	.7331E+03	.8416
9	.4745E+04 (MAX)	.6665E+03	.3492E+03	.4503E+03	.7739
10	.4788E+04 (MAX)	.1535E+04	.1606E+04	.1090E+04	.9991
10	.4299E+04 (MED)	.1535E+04	.1069E+04	.1172E+04	.9955
10	.4788E+04 (MAX)	.9520E+03	.9622E+03	.6547E+03	.9768
10	.4299E+04 (MED)	.9520E+03	.4147E+03	.7098E+03	.8788
10	.4788E+04 (MAX)	.5437E+03	.3856E+02	.3565E+03	0.0000
11	.4385E+04 (MAX)	.3050E+04	.1914E+04	.2377E+04	1.0000
11	.3124E+04 (MED)	.3050E+04	.5336E+03	.2697E+04	.9924
11	.4385E+04 (MAX)	.2634E+04	.1770E+04	.2039E+04	1.0000
11	.3124E+04 (MED)	.2634E+04	.3828E+03	.2322E+04	.9714
11	.4385E+04 (MAX)	.2135E+04	.1549E+04	.1637E+04	.9998
11	.3124E+04 (MED)	.2135E+04	.1513E+03	.1874E+04	0.0000
11	.4385E+04 (MAX)	.1634E+04	.1241E+04	.1237E+04	.9979
11	.4385E+04 (MAX)	.1525E+04	.1156E+04	.1150E+04	.9965
11	.4385E+04 (MAX)	.8444E+03	.3260E+03	.6171E+03	.8095

MEAN PI/H = .7698E+00

FIGURE 7-1. SAMPLE OUTPUT OF PROGRAM SECTION IV

```
COLUMN          111111111122222222223333333333444444444455555555556666666666777777778
NUMBER   1234567890123456789012345678901234567890123456789012345678901234567890123456789
 CARD
NUMBER
   1     SECTION IV
                   *** THE FOLLOWING CARDS DEFINE ONE BATCH OF CASES
                       THE VARIOUS CARD SETS MAY OCCUR IN ANY ORDER ***
                                      *** CARD SET A ***
           *** THIS SET OF CARDS IS IDENTICAL WITH CARDS 1 THRU 14 OF FIGURE 4-9 ***
                                      *** CARD SET B ***
   B     TERMINAL VELOCITY   VTERM
                                      *** CARD SET C ***
  C1     THOR DATA NTHOR
  C2     METAL(I)  CV(I)     ALFAV(I)  BETAV(I)  GAMAV(I)  LAMDAV(I) KFCTRV(I)
  C3               CM(I)     ALFAM(I)  BETAM(I)  GAMAM(I)  LAMDAM(I) KFCTRM(I)
                    *** UP TO 19 MORE PAIRS OF CARDS C2 AND C3, AS
                        NEEDED, FOR AS MANY AS 20 SETS OF THOR DATA ***
                                      *** CARD SET D ***
   D     ANGLES    NTHETA    THETA(1)  THETA(2)  ...
            *** UP TO 2 MORE CARDS, AS NEEDED, FOR AS MANY AS 20 VALUES OF THETA ***
                                      *** CARD SET E ***
  E1     ARMOR     NPLATE
  E2     PLATE(1)  T(1)      PLATE(2)  T(2)       ...
                   *** UP TO 4 MORE CARDS, AS NEEDED, FOR UP TO 20 PLATES ***
                                      *** CARD SET F ***
   F     INCAPACITATION PARAMETERS     SUMMER    COEF1     COEF2     COEF3
                                   *** LAST CARD OF THE BATCH ***
 LAST    COMPUTE
```

FIGURE 7-2. DATA DECK FOR PROGRAM SECTION IV

7.3.2 Definition of Each Input Variable.

Card 1:

Columns 1-10 - the words SECTION IV.

The following card sets define a batch of cases. The card sets may appear in any order. The batch is terminated by a COMPUTE card. As many batches as desired may follow the SECTION IV card.

Card set A:

This card set inputs the fragmentation data and is identical with cards 11 through 14 of paragraph 4.3.2.

Card set B:

Columns 1-10 - the words TERMINAL V.

VTERM - the projectile terminal velocity in feet per second.

Card set C:

This card set inputs as many as 20 sets of Thor constants from which are drawn the appropriate sets for the various armor plates or armor plate assemblies in this batch.

Card C1:

Columns 1-10 - the words THOR DATA.

NTHOR - the number of sets of Thor constants; $1. \leq \text{NTHOR} \leq 20.$

Cards C2 and C3:

This card pair defines one set of Thor constants. The cards are identical with cards 8 and 9 of paragraph 4.3.2. There must be NTHOR such card pairs.

Card set D:

Columns 1-10 - the word ANGLES.

NTHETA - the number of obliquity angles to be considered for each armor plate or plate assembly; $1. \leq \text{NTHETA} \leq 20.$

THETA(I) - the I^{th} obliquity angle to be considered for each armor plate or plate assembly; in degrees; there must be NTHETA such angles; $0.° \leq \text{THETA} \leq 90°.$

Card set E:

This card set defines the armor plates processed in this batch.

Card E1:

Columns 1-10 - the word ARMOR.

NPLATE - the number of plates.

Card E2:

This card defines an armor plate. Exactly NPLATE such plates must be defined.

PLATE(I) - 10 alphanumeric characters; specifies the material of which the I^{th} plate is made; must match, character by character, one of the variables METAL(K) from card set C.

T(I) - the thickness, in inches, of the I^{th} plate.

Card set F:

This card set defines the incapacitation parameters used in computing $P_{I/H}$.

Columns 1-10 - the word INCAPACITA.

Columns 31-40 - the word SUMMER or the word WINTER; describes the type of uniform being worn.

COEF1, COEF2, COEF3 - three constants used to compute $P_{I/H}$ for the selected kill criterion (see reference 5 and appendix B).

Last card of any batch:

Columns 1-10 - the word COMPUTE.

8. THE COMPLETE JOB DECK

8.1 The Data Deck.

Thus far, the four sections of the program and their data decks (herein called subdecks) have been discussed separately. In an actual machine run, any number of each of the four types of data subdecks may be stacked in any order desired, so long as one insures that, when a subdeck is processed, it will not demand information produced by some later subdeck.

Each type of subdeck has certain options allowing the use of data read in from a preceding subdeck of the same type. An important constraint upon the complete data deck is that each of the four program sections destroys the input data belonging to any previous different program section, with one exception, which is discussed below.

Program sections I and IV do not destroy each other's static fragmentation data. That is, either of these program sections can use static fragmentation data read in by itself or by the other and will leave that data ready to be used again by itself or by the other. Any time either of these program sections reads in such data, it replaces the fragmentation data previously read in by either section.

A complete data deck must contain at its end either of two cards not heretofore discussed.

One of these is a card with the words END OF DAT in columns 1 through 10. This card terminates the data deck. Succeeding cards are ignored.

The other card can have any of the forms shown in Figure 8-1. Only columns 1-10 and 41-50 are read; other columns are ignored. These cards cause termination of the run immediately after completion of the specified operations.

EXPECTED P causes an index of the expected perforations matrix file to be printed. An example is at Figure C-13. The column headings have the following meanings:

FILES is the unique integer identifying an expected perforations matrix; used as an element of the set FILE (paragraph 5.4.2, card 5).

NXX, NYY are as in paragraph 4.3.2, card 1.

CELL is the edge length of the cell with which the ground plane is gridded, in inches.

HOB is the height of burst above the ground, in meters.

IROW, ICOL are as in paragraph 4.3.2, card 1.

```
COLUMN            1111111111222222222233333333334444444444555555555566666666667777777777
NUMBER   12345678901234567890123456789012345678901234567890123456789012345678901234567890

         EXPECTED PERFORATIONS FILE              DAMAGE FUNCTION FILE
         DAMAGE FUNCTION FILE                    EXPECTED PERFORATIONS FILE
         EXPECTED PERFORATIONS FILE
         DAMAGE FUNCTION FILE
```

FIGURE 8-1. OPTIONAL CARD

LYMYB, LYMYT, LYMXL, LYMXR are the row and column numbers which define the minimum and maximum deflection and range limits, respectively, of the non-zero portion of the expected perforations matrix. If the matrix is all zeroes, then these numbers merely define a small region about the projectile.

A, B, NSIDE, NX, NY, NZ, T are elements of the sets ALPHA, BETA, NSIDE, NXTGT, NYTGT, NZTGT, and TT, respectively (paragraph 4.3.2, card 5).

THCK is an element of the set TTHCK (paragraph 4.3.2, card 3).

DAMAGE FUN causes the contents of the damage function file to be printed. An example is at Figure C-14. The column headings have the following meanings:

FILE NUMBER is the unique integer identifying a damage function; used as FILE2 (paragraph 6.4.2, card 14).

POINT NUMBER identifies the I^{th} case of a batch of cases from program section II.

SYMMETRY CHARACTER is KSYM (paragraph 6.4.2, card 1).

DAMAGE CRITERION is the number of perforations defined as the damage criterion.

LYMXL, LYMXR, LYMYB, LYMYT are as above.

PROJECTILE LOCATION is the range, deflection coordinate pair of the burst point with respect to the probability of damage function.

HOB is the height of burst above the ground, in meters.

CELL is as above.

NUMBER OF REGIONS is the number of different valued probability of damage regions in the probability of damage function.

THE DAMAGE FUNCTION REGIONS data completely defines the unnormalized damage function.

8.2 The Control Deck.

Certain control cards are needed to cause problem loading and execution and to insure availability and permanence of files. Depending on the situation, these functions can be accomplished in any number of

ways, and hence will not be discussed in detail. The control decks used to produce the example of Appendix C are presented and explained in Appendix D. One point, however, is worthy of note. On several figures providing examples of output in the body of the report and in Appendix C, the names assigned by the user to his perforation matrix and damage function files appear. These names are made available to the program by the control deck. Note from Appendix A that the first FORTRAN statement in the program is:

```
PROGRAM LASVEM (INPUT, OUTPUT, TAPE1, TAPE2,...)
```

where:

INPUT is the system input file.
OUTPUT is the system output file.
TAPE1 is the perforation matrix file.
TAPE2 is the damage function file.

The user's local file names for the perforation matrix file and the damage function file must be equivalenced, by appropriate commands, to TAPE1 and TAPE2, respectively.

BIBLIOGRAPHY

1. Institute for Cooperative Research, Ballistic Analysis Laboratory; The Resistance of Various Metallic Materials to Perforation by Steel Fragments; Empirical Relationships for Fragment Residual Velocity and Residual Weight (U); Project Thor Technical Report No. 47; The Johns Hopkins University, Baltimore, Maryland, April 1961; UNCLASSIFIED.

2. Institute for Cooperative Research, Ballistic Analysis Laboratory; The Resistance of Aluminum Alloy Targets to Perforation by Small-Caliber, Armor-Piercing Projectiles (U); Project Thor Technical Report No. 70; The Johns Hopkins University, Baltimore, Maryland, January 1969; CONFIDENTIAL.

3. Myers, K. A., Lethal Area Description; BRL Technical Note No. 1510; Aberdeen Proving Ground, Maryland, July 1963; UNCLASSIFIED.

4. Snow, Roger and Ryan, Margaret; A Simplified Weapons Evaluation Model; US Air Force Project Rand Memorandum RM-5677-1-PR; The Rand Corporation, Santa Monica, California, May 1970; UNCLASSIFIED.

5. Kokinakis, William and Sperazza, Joseph; Criteria for Incapacitating Soldiers with Fragments and Flechettes (U); BRL Report No. 1269; Aberdeen Proving Ground, Maryland, January 1965; CONFIDENTIAL.

6. Development and Proof Services; Firing Record No. B14449; Aberdeen Proving Ground, Maryland, April 21, 1961; CONFIDENTIAL.

7. Heppner, L. D., Fragmentation Test Design, Collection, Reduction and Analysis of Data; APG Miscellaneous Report No. 306; Aberdeen Proving Ground, Maryland; UNCLASSIFIED.

8. Disposition Form, USABRL (AMXRD-BVL), 27 Aug 69, subject: Ballistic Limits for Fragments (U); CONFIDENTIAL.

The following page is blank.

APPENDIX A

PROGRAM TABULATION

The Following Page is Blank.

```
      PROGRAM LASVEM (INPUT,OUTPUT,TAPE1,TAPE2,TAPE10,TAPE5=INPUT,       *   0* 0  000100
     1  TAPE6=OUTPUT)                                                    LASVM 2  000110
C                                                                        LASVM 3  000120
C                                                                        LASVM 4  000130
C     AN OVERLAID STORAGE SCHEME IS USED IN LABELLED COMMON. THEREFORE,  LASVM 5  000140
C     CAUTION MUST BE EXERCISED WHEN CHANGING COMMON AND WHEN CHANGING   LASVM 6  000150
C     THE COMPUTATION SEQUENCE. COM1 IS 20100 WORDS USED PRIMARILY AS    LASVM 7  000160
C     VARIOUS (100,100), (100,200), (10000), AND (20000) WORD ARRAYS.    LASVM 8  000170
C     COM3 IS 10000 WORDS USED AS A TEMPORARY (100,100) WORD ARRAY AND   LASVM 9  000180
C     AS STORAGE FOR VARIOUS OTHER ARRAYS AND VARIABLES.                 LASVM10  000190
C                                                                        LASVM11  000200
      COMMON /COM1/COM1(20100)                                           LASVM12  000210
      DIMENSION LAMDA2(100,200),PERF1(100,100),LAMDA(100,100)            LASVM13  000220
     1          ,ABUFFR(10100),BUFFER(10100)                             LASVM14  000230
      EQUIVALENCE (COM1(1),LAMDA2(1,1),PERF1(1,1))                       LASVM15  000240
     1            ,(COM1(10001),LAMDA(1,1),ABUFFR(1),BUFFER(1))          LASVM16  000250
C                                                                        LASVM17  000260
      COMMON /COM3/COM3(10000)                                           LASVM18  000270
      DIMENSION PERF2(100,100),VDYN(92),NSP(46),M(40,46),SGST(46)        LASVM19  000280
     1          ,HHHOB(25),NXTGT(25),FNXTGT(25),NYTGT(25),FNYTGT(25)     LASVM20  000290
     2          ,NZTGT(25),FNZTGT(25),NSIDE(25),FNSIDE(25),ALPHA(25)     LASVM21  000300
     3          ,BETA(25),TT(25),TTHCK(25)                               LASVM22  000310
      EQUIVALENCE (COM3(1),PERF2(1,1)),(COM3(231),VDYN(1))               LASVM23  000320
     1            ,(COM3(323),NSP(1)),(COM3(369),M(1,1))                 LASVM24  000330
     2            ,(COM3(4049),SGST(1)),(COM3(4187),HHHOB(1))            LASVM25  000340
     3            ,(COM3(4212),NXTGT(1),FNXTGT(1))                       LASVM26  000350
     4            ,(COM3(4237),NYTGT(1),FNYTGT(1))                       LASVM27  000360
     5            ,(COM3(4262),NZTGT(1),FNZTGT(1))                       LASVM28  000370
     6            ,(COM3(4287),NSIDE(1),FNSIDE(1))                       LASVM29  000380
     7            ,(COM3(4312),ALPHA(1)),(COM3(4337),BETA(1))            LASVM30  000390
     8            ,(COM3(4362),TT(1)),(COM3(4387),TTHCK(1))              LASVM31  000400
     9            ,(COM3(9936),BLAST),(COM3(9937),FRGFLG)                LASVM32  000410
     A            ,(COM3(9940),SHAPE),(COM3(9941),AA),(COM3(9942),NNSP)  LASVM33  000420
     B            ,(COM3(9943),CV),(COM3(9944),ALFAV),(COM3(9945),BETAV)LASVM34  000430
     C            ,(COM3(9946),GAMAV),(COM3(9947),LAMDAV)                LASVM35  000440
     D            ,(COM3(9948),CM),(COM3(9949),ALFAM),(COM3(9950),BETAM)LASVM36  000450
     E            ,(COM3(9951),GAMAM),(COM3(9952),LAMDAM)                LASVM37  000460
     F            ,(COM3(9953),CNST1),(COM3(9954),CNST3V)                LASVM38  000470
     G            ,(COM3(9955),CNST4V),(COM3(9957),CNST3M)               LASVM39  000480
     H            ,(COM3(9958),CNST4M),(COM3(9960),EPSLN)                LASVM40  000490
      EQUIVALENCE  (COM3(9961),VTERM),(COM3(9962),W)                     LASVM41  000500
     1            ,(COM3(9963),NTRGT,FNTRGT),(COM3(9964),ITRGT)          LASVM42  000510
     2            ,(COM3(9965),NX),(COM3(9966),NY),(COM3(9967),NZ)       LASVM43  000520
     3            ,(COM3(9968),NSID),(COM3(9969),A),(COM3(9970),B)       LASVM44  000530
     4            ,(COM3(9971),T),(COM3(9972),NTHCK,FNTHCK)              LASVM45  000540
     5            ,(COM3(9973),ITHCK),(COM3(9974),THCK)                  LASVM46  000550
     6            ,(COM3(9975),IKRIT),(COM3(9977),SHELX)                 LASVM47  000560
     7            ,(COM3(9978),SHELY),(COM3(9979),LIMXL)                 LASVM48  000570
     8            ,(COM3(9980),LIMXR),(COM3(9981),LIMYB)                 LASVM49  000580
     9            ,(COM3(9982),LIMYT),(COM3(9983),K22),(COM3(9984),K222)LASVM50  000590
     A            ,(COM3(9985),H),(COM3(9986),H2),(COM3(9987),SINW)      LASVM51  000600
     B            ,(COM3(9988),COSW),(COM3(9989),HSINW)                  LASVM52  000610
     C            ,(COM3(9990),SINA),(COM3(9991),COSA),(COM3(9992),TANA)LASVM53  000620
     D            ,(COM3(9998),IFLAG)                                    LASVM54  000630
C                                                                        LASVM55  000640
      COMMON /DISC/DISC(15016)                                           LASVM56  000650
      DIMENSION PLIST(2502,2),SLIST(5002,2)                              LASVM57  000660
      EQUIVALENCE (DISC(1),PLIST(1,1)),(DISC(5005),SLIST(1,1))           LASVM58  000670
     1            ,(DISC(15009),FILES,FFILES),(DISC(15010),NFILE,FNFILE) LASVM59  000680
     2            ,(DISC(15011),NSTAPS,FNSTAP),(DISC(15012),PTAPE1)      LASVM60  000690
```

```
     3         ,(DISC(15013),STAPE) ,(DISC(15014),TAPE)
     4         ,(DISC(15015),TNAME) ,(DISC(15016),SNAME)
C
      COMMON /AAA/AAA(6)
      COMMON /CELL/CELL
      COMMON /COLROW/ICOL,IROW,ICOL1,IROW1
      EQUIVALENCE (ICOL,FICOL),(IROW,FIROW)
      COMMON /CONSTT/   TWOPI,RTODG,DGTOR,FTTOM,DIMX,DIMY
      COMMON /HOB/IHOB,NHOB,HOB
      EQUIVALENCE (IHOB,IPNT),(NHOB,FNHOB,NPNTS,FNPNTS)
      COMMON /KRIT/KRIT(7)
      DIMENSION FKRIT(7)
      EQUIVALENCE (KRIT(1),FKRIT(1))
      COMMON /KSYM/KSYM
      EQUIVALENCE (FKSYM,KSYM)
      COMMON /LOMIT/LOMXL,LOMXR
      COMMON /LYMIT/LYMXL,LYMXR,LYMYB,LYMYT
      COMMON /NOPK/NOPK
      COMMON /NXXNYY/NXX,NYY
      EQUIVALENCE (NXX,FNXX),(NYY,FNYY)
      COMMON /PAGE/LINES,PAGE
      COMMON /SHIFTS/FILE(7),SHFTX(7),SHFTY(7)
      DIMENSION FFILE(7),FSHFTX(7),FSHFTY(7)
      EQUIVALENCE (FILE(1),FFILE(1)),(SHFTX(1),FSHFTX(1))
     1         ,(SHFTY(1),FSHFTY(1))
      COMMON /SHOW/SHOW(25)
      DIMENSION FSHOW(25)
      EQUIVALENCE (SHOW(1),FSHOW(1))
      COMMON /THOR7C/RATIOM,RATIOV
      COMMON /VOID2/VOID2
C
      DIMENSION IFMT1(15),IFMT2(15),SCALE1(25),WORD(6)
      EQUIVALENCE (PI,HLFPI),(NKRIT,FNKRIT)
C
      INTEGER AAA,BUFFER,CCC,DELTX,DELTY,DIMX,DIMY,FILE
     1         ,FILES,FRGFLG,PAGE,PLIST,PTAPE1,SHFTX,SHFTY
     2         ,SHO,SHOW,SLIPX,SLIPY,SLIST,SNAME,SNUM1,SNUM2,STAPE,TAPE
     3         ,TNAME,VOID2,WORD
      REAL KFCTRM,KFCTRV,LAMDA,LAMDAM,LAMDAV,LAMDA2,M,MTOFT
C
      DATA (IFMT1(I),I=1,12)/10H(1H1/  34H,10H PROBABILI,10HTY OF SUST
     1         ,10HAINING AT ,10HLEAST,I3,1,10H3H PERFORA,10HTIONS,
     2         ,10H9X,5HHOB =,10H,F6.2,7H M,10HETERS,9X,6,10HHCELL =,F5
     3         ,10H.2,7H INCH/
      DATA (IFMT2(I),I=1,15)/10H(/54H THE ,10HTARGET/HOB,10H COMBINATI
     1         ,10HON WAS ASS,10HEMBLED FRO,10HM RECORDS,,10H  (I5,1H,)
     2         ,10H,4H AND,I5,10H/2(/1H ,2E,10HX,10HTHE C,10HENTER,A3,1
     3         ,10H2HOFFSETS ,10HWERE,  (I5,10H,1H,),4H A,10HND,I5/))  /
      DATA (WORD(I),I=1,6)  /10H COLUMN   ,10H ROW      ,10H MIGHT
     1         ,10H WOULD    ,10H X        ,10H Y        /
C
C
      PAGE=60
C     PAGE IS THE NUMBER OF LINES OF OUTPUT PER PAGE
      DIMX=100
      DIMY=100
C     DIMX AND DIMY PROVIDE DIMENSION INFORMATION ON LARGE ARRAYS. THEY
C     MUST AGREE WITH THE APPROPRIATE COMMON AND DIMENSION STATEMENTS.
C     DIMX AND DIMY WILL CAUSE OUTPUT TROUBLE IF THEY GET TOO BIG. DIMX
C     AND DIMY ARE SIMILAR TO NXX AND NYY, WHICH ARE USER ORIENTED. DIMX
```

```
C     AND DIMY ARE PROGRAMMER ORIENTED AND SERVE AS ACHIEVABLE UPPER
C     BOUNDS FOR NXX AND NYY.
C
      TAPE=1
      STAPE=2
      PTAPE1=10
C     TAPE IS THE PERFORATION MATRIX FILE.
C     STAPE IS THE DAMAGE FUNCTION FILE.
C     PTAPE1 IS A SCRATCH FILE.
C     PROGRAM LOGIC ORIGINALLY ASSUMED SEQUENTIAL FILES ON TAPE. MINIMAL
C     LOGIC CHANGES HAVE BEEN MADE TO GET ON THE CDC MACHINE. THEREFORE,
C     RECORDS WITHIN A LOGICAL INPUT DATA SET ARE STILL GENERATED
C     SEQUENTIALLY, EVEN THOUGH THEY ARE STORED AND ACCESSED RANDOMLY.
C     THAT SET OF RECORDS CAN BE LOCATED, AS A SET, ANYWHERE WITHIN THE
C     FILE. WITHIN THE SET, LOGICAL RECORDS ARE IN SEQUENCE. THIS
C     APPLIES TO BOTH DAMAGE FUNCTIONS AND PERFORATION MATRICES. PTAPE1
C     IS STILL A SEQUENTIAL FILE.
      CALL OPENMS (TAPE,PLIST,2502,0)
      CALL OPENMS (STAPE,SLIST,5002,0)
      DO 10 I=1,2502
      IF (PLIST(I,1).NE.0) GO TO 20
   10 CONTINUE
      GO TO 30
   20 CALL READMS (TAPE,PLIST(1,2),2501,2501)
   30 DO 40 I=1,5002
      IF (SLIST(I,1).NE.0) GO TO 50
   40 CONTINUE
      GO TO 60
   50 CALL READMS (STAPE,SLIST(1,2),5001,5001)
   60 CALL FILNAM (TNAME,SNAME)
C
      PI=3.1415926536
      TWOPI=2.*PI
      HLFPI=PI/2.
      MTOFT=3.28083
      FTTOM=1./MTOFT
      RTODG=57.2957795131
      DGTOR=1./RTODG
      ENAT=2.71828182846
      VCID2=0
   70 READ (5,1090) (AAA(I),I=1,8)
      IF (AAA(1).EQ.10HASSEMBLE  .OR.AAA(1).EQ.10H ASSEMBLE .OR.AAA(1).E
     1Q.10H  ASSEMBLE) GO TO 520
   80 IF (AAA(1).EQ.10HEXPECTED P.OR.AAA(1).EQ.10HDAMAGE FUN) CALL INDEX
     1 (0)
      IF (AAA(1).NE.10HENCOUNTER .AND.AAA(1).NE.10H ENCOUNTER) GO TO 90
      CALL SNOW
      GO TO 70
   90 IF (AAA(1).NE.10HSECTION IV) GO TO 100
      CALL SEXUN4
      GO TO 70
  100 IF (AAA(1).EQ.10HEND OF DAT) GO TO 110
      IF (AAA(1).EQ.10HBASIC DATA) GO TO 120
      WRITE (6,1350) (AAA(I),I=1,8)
  110 CALL WRITMS (TAPE,PLIST(1,2),2501,2501,-1,0)
      CALL WRITMS (STAPE,SLIST(1,2),5001,5001,-1,0)
      CALL CLOSMS (TAPE)
      CALL CLOSMS (STAPE)
      WRITE (6,1360)
      STOP
```

```
C                                                                       LASV181  001900
C     THIS PART OF MAIN COMPUTES THE PERFORATIONS SUSTAINED BY COMPONENTLASV182  001910
C     PLATES OF THE TARGET                                              LASV183  001920
C                                                                       LASV184  001930
  120 KSYM=0                                                            LASV185  001940
C     METHODOLOGY DATA                                                  LASV186  001950
C     THIS LOGICAL DATA DECK GENERATES SEQUENTIALLY A SET OF PERFORATIONLASV187  001960
C     MATRICES WHICH WILL BE STORED SEQUENTIALLY IN A RANDOM ACCESS FILELASV188  001970
C     (TAPE=TAPE1=1). THE INPUT VALUE OF FILES IS THE KEY FOR THE 1ST   LASV189  001980
C     LOGICAL RECORD (MATRIX) IN THE SET. THE FILE INFORMATION TABLE    LASV190  001990
C     (PLIST) ALLOWS NOT MORE THAN 2500 LOGICAL RECORDS WITH KEYS 1 THRULASV191  002000
C     2500. SIMILARLY, DAMAGE FUNCTIONS ARE GENERATED SEQUENTIALLY      LASV192  002010
C     WITHIN A SET RANDOMLY LOCATED ON A RANDOM ACCESS FILE (STAPE=TAPE2LASV193  002020
C     =2). THE FILE INFORMATION TABLE (SLIST) ALLOWS NOT MORE THAN 5000 LASV194  002030
C     LOGICAL RECORDS WITH KEYS 1 THRU 5000. THE INPUT VALUE OF NSTAPE  LASV195  002040
C     IS THE KEY FOR THE 1ST LOGICAL RECORD (DAMAGE FUNCTION) IN THE SETLASV196  002050
      DECODE (60,1110,AAA(2) )FFILES,CELL1,FNXX,FNYY,FICOL,FIROW        LASV197* 002060
      FILES=FFILES+.1                                                   LASV198  002070
      NXX=FNXX+.1                                                       LASV199  002080
      NYY=FNYY+.1                                                       LASV200  002090
      ICOL=FICOL+SIGN(.1,FICOL)                                         LASV201  002100
      IROW=FIROW+SIGN(.1,FIROW)                                         LASV202  002110
      IROW1=IROW                                                        LASV203  002120
      ICOL1=ICOL                                                        LASV204  002130
      IF (FILES.GT.0) GO TO 150                                         LASV205  002140
      DO 130 I=1,2500                                                   LASV206  002150
      J=2501-I                                                          LASV207  002160
      IF (PLIST(J,2).NE.0) GO TO 140                                    LASV208  002170
  130 CONTINUE                                                          LASV209  002180
      J=0                                                               LASV210  002190
  140 FILES=J+1                                                         LASV211  002200
C     TARGET DATA                                                       LASV212  002210
  150 READ (5,1110) FNHOB,(HHHOB(I),I=1,7)                              LASV213R 002220
      NHOB=FNHOB+.1                                                     LASV214  002230
      IF (NHOB.GT.7) READ (5,1110) (HHHOB(I),I=8,NHOB)                  LASV215R 002240
      READ (5,1110) FNTHCK,(TTHCK(I),I=1,7)                             LASV216R 002250
      NTHCK=FNTHCK+.1                                                   LASV217  002260
      IF (NTHCK.GT.7) READ (5,1110) (TTHCK(I),I=8,NTHCK)                LASV218R 002270
      READ (5,1110) FNTRGT                                              LASV219R 002280
      NTRGT=FNTRGT+.1                                                   LASV220  002290
      READ (5,1130) (FNXTGT(I),FNYTGT(I),FNZTGT(I),FNSIDE(I),ALPHA(I),  LASV221R 002300
     1  BETA(I),TT(I),SCALE1(I),FSHOW(I),I=1,NTRGT)                     LASV222R 002310
      DO 160 I=1,NTRGT                                                  LASV223  002320
      NXTGT(I)=FNXTGT(I)+.1                                             LASV224  002330
      NYTGT(I)=FNYTGT(I)+.1                                             LASV225  002340
      NZTGT(I)=FNZTGT(I)+.1                                             LASV226  002350
      NSIDE(I)=FNSIDE(I)+.1                                             LASV227  002360
  160 SHOW(I)=FSHOW(I)+.1                                               LASV228  002370
C     SCALE1 IS A FACTOR USED TO ADJUST FOR SLIGHT MISMATCH IN SURFACE  LASV229  002380
C     AREA BETWEEN THE ACTUAL AND MODEL PLATES. EFFECTIVE ONLY ON THE   LASV230  002390
C     FRAGMENT CONTRIBUTIONS TO THE PERFORATION MATRICES.               LASV231  002400
C     SCALE1 = INPUT PLATE SURFACE AREA / TRUE PLATE SURFACE AREA       LASV232  002410
      READ (5,1110) FNKRIT,(FKRIT(IKRIT),IKRIT=1,7)                     LASV233R 002420
      NKRIT=FNKRIT+.1                                                   LASV234  002430
      DO 170 I=1,NKRIT                                                  LASV235  002440
  170 KRIT(I)=FKRIT(I)+.1                                               LASV236  002450
C     THE KRIT ARRAY MUST BE MONOTONE INCREASING                        LASV237  002460
      READ (5,1090) (AAA(I),I=1,8)                                      LASV238R 002470
C     PHYSICAL CONSTANTS OF THE PLATE                                   LASV239  002480
      IF (AAA(1).NE.10HTHOR DATA .AND.AAA(1).NE.10H THCK DATA) GO TO 180LASV240  002490
```

```
      READ (5,1120) METAL,CV,ALFAV,BETAV,GAMAV,LAMDAV,KFCTRV            LASV241R 002500
      READ (5,1110) CM,CM,ALFAM,BETAM,GAMAM,LAMDAM,KFCTRM               LASV242R 002510
C     FROM THGR 7C, KFCTRM(2024)=.964 AND KFCTRV(2024)=.965             LASV243  002520
      RATIOM=0.                                                         LASV244  002530
      RATIOV=C.                                                         LASV245  002540
      IF (KFCTRM.NE.0.) RATIOM=ALOG10(KFCTRM/.964)                      LASV246  002550
      IF (KFCTRV.NE.0.) RATIOV=ALOG1C(KFCTRV/.965)                      LASV247  002560
      SAVE1=GAMAM                                                       LASV248  002570
      SAVE2=GAMAV                                                       LASV249  002580
      SAVE3=LAMDAM                                                      LASV250  002590
      READ (5,1090) (AAA(I),I=1,8)                                      LASV251R 002600
      GO TO 190                                                         LASV252  002610
  180 GAMAM=SAVE1                                                       LASV253  002620
      GAMAV=SAVE2                                                       LASV254  002630
      LAMDAM=SAVE3                                                      LASV255  002640
C     PROJECTILE LOCATION IS SHELX, SHELY                               LASV256  002650
  190 SHELX=FLOAT(ICOL)+.5                                              LASV257  002660
      SHELY=FLOAT(IROW)+.5                                              LASV258  002670
      CELL=CELL1/12.                                                    LASV259  002680
C     PROJECTILE DATA                                                   LASV260  002690
      DECODE (30,111C,AAA(1) )W,VTERM,BLAST                             LASV261* 002700
      READ (5,1090) (AAA(I),I=1,8)                                      LASV262R 002710
      IF (AAA(1).NE.10HFRAGMENTAT) GO TO 200                            LASV263  002720
      FRGFLG=0                                                          LASV264  002730
      CALL FRAG1 (1H1)                                                  LASV265  002740
  200 EPSLN=2.*BLAST/CELL1                                              LASV266  002750
C     EPSLN IS IN HALF-CELLS                                            LASV267  002760
      CNST1=-AA*CELL/2.*ALOG10(ENAT)                                    LASV268  002770
      TEMP3=2.*ALOG10(SHAPE)/3.                                         LASV269  002780
      CNST2V=CV-ALFAV*TEMP3                                             LASV270  002790
      CNST2M=CM-ALFAM*TEMP3                                             LASV271  002800
      CNST3V=(2.*ALFAV/3.+BETAV)/(1.-LAMDAV)                            LASV272  002810
      CNST3M=2.*ALFAM/3.+BETAM                                          LASV273  002820
      WRAD=W*DGTOR                                                      LASV274  002830
      COSW=COS(WRAD)                                                    LASV275  002840
      SINW=SIN(WRAD)                                                    LASV276  002850
C     OUTPUT THE INPUT                                                  LASV277  002860
      WRITE (6,1140) METAL,CV,ALFAV,BETAV,GAMAV,LAMDAV,KFCTRV,CM,ALFAM, LASV278W 002870
     1  BETAM,GAMAM,LAMDAM,KFCTRM,W,VTERM,BLAST,SHAPE,AA                LASV279W 002880
      GAMAV=GAMAV/(1.-LAMDAV)                                           LASV280  002890
      LAMDAM=LAMDAM/(1.-CNST3M)                                         LASV281  002900
      GAMAM=GAMAM/(1.-CNST3M)                                           LASV282  002910
C                                                                       LASV283  002920
      IF (AAA(1).NE.10HFRAGMENTAT) GO TO 210                            LASV284  002930
      CALL FRAG2 (1H1)                                                  LASV285  002940
      GO TO 220                                                         LASV286  002950
  210 WRITE (6,1340)                                                    LASV287W 002960
C                                                                       LASV288  002970
  220 CALL VECTR                                                        LASV289  002980
C                                                                       LASV290  002990
      DO 260 K2=K22,K222                                                LASV291  003000
      IF (SGST(K2).EQ.0.) GO TO 260                                     LASV292  003010
      IF (AAA(1).EQ.10HFRAGMENTAT) GO TO 230                            LASV293  003020
      IF (FRGFLG.EQ.0) GO TO 250                                        LASV294  003030
  230 NNSP=NSF(K2)                                                      LASV295  003040
      DO 240 K1=1,NNSP                                                  LASV296  003050
  240 M(K1,K2)=ALOG10(M(K1,K2))                                         LASV297  003060
  250 VDYN(K2)=ALOG1C(VDYN(K2))                                         LASV298  003070
      VDYN(K2+46)=ALOG10(VDYN(K2+46))                                   LASV299  003080
  260 CONTINUE                                                          LASV300  003090
```

```
      FRGFLG=0                                                          LASV301   003100
      CCC=0                                                             LASV302   003110
      NOPK=0                                                            LASV303   003120
      IF (AAA(1).EQ.10HFRAGMENTAT) READ (5,1090) (AAA(I),I=1,6)         LASV304R  003130
      IF (AAA(1).NE.10HCONTOUR    .AND.AAA(1).NE.10H CONTOUR  .AND.AAA(1)LASV305   003140
     1.NE.10H  CONTOUR .AND.AAA(1).NE.10H   CONTOUR) GO TO 270          LASV306   003150
      DECODE (10,1110,AAA(2) )FNSTAP                                    LASV307*  003160
      GO TO 280                                                         LASV308   003170
  270 IF (AAA(4).NE.10HCONTOUR    .AND.AAA(4).NE.10H CONTOUR  .AND.AAA(4)LASV309   003180
     1.NE.10H  CONTOUR .AND.AAA(4).NE.10H   CONTOUR) GO TO 310          LASV310   003190
      DECODE (10,1110,AAA(5) )FNSTAP                                    LASV311*  003200
  280 NSTAPE=FNSTAP+.1                                                  LASV312   003210
      CCC=10HCONTOUR                                                    LASV313   003220
      IF (NSTAPE.GT.0) GO TO 310                                        LASV314   003230
      DO 290 I=1,5000                                                   LASV315   003240
      J=5001-I                                                          LASV316   003250
      IF (SLIST(J,2).NE.0) GO TO 300                                    LASV317   003260
  290 CONTINUE                                                          LASV318   003270
      J=0                                                               LASV319   003280
  300 NSTAPE=J+1                                                        LASV320   003290
  310 IF (AAA(1).EQ.10HNO PROBABI.OR.AAA(3).EQ.10HNO PROBABI) NOPK=10HNOLASV321   003300
     1 PROBABI                                                          LASV322   003310
      ENCODE (10,1250,IFMT1(13) )                                       LASV323*  003320
      ENCODE (70,1260,IFMT1(1) )                                        LASV324*  003330
C                                                                       LASV325   003340
C     START THE SOLUTION.  LOOP ON BURST HEIGHT. THROUGHOUT THE         LASV326   003350
C     COMPUTATION, THE UNIT OF LENGTH IS A HALF CELL, EXCEPT FOR THE    LASV327   003360
C     PLATE THICKNESS.                                                  LASV328   003370
C     NHOB*NTHCK*NTRGT MUST BE LESS THAN 2501                           LASV329   003380
      DO 510 IHOB=1,NHOB                                                LASV330   003390
      HOB=HHHOB(IHOB)                                                   LASV331   003400
C                                                                       LASV332   003410
C     LOOP ON PLATE THICKNESS                                           LASV333   003420
      DO 510 ITHCK=1,NTHCK                                              LASV334   003430
      THCK=TTHCK(ITHCK)                                                 LASV335   003440
      CNST4M=ALOG10(THCK)                                               LASV336   003450
      CNST4V=(CNST2V+ALFAV*CNST4M)/(1.-LAMDAV)                          LASV337   003460
      CNST4M=(CNST2M+ALFAM*CNST4M)/(1.-CNST3M)                          LASV338   003470
      CALL GOTHRU                                                       LASV339   003480
C                                                                       LASV340   003490
C     LOOP ON TARGETS                                                   LASV341   003500
      DO 510 ITRGT=1,NTRGT                                              LASV342   003510
      NX=NXTGT(ITRGT)                                                   LASV343   003520
      NY=NYTGT(ITRGT)                                                   LASV344   003530
      NZ=NZTGT(ITRGT)                                                   LASV345   003540
      T=TT(ITRGT)*FTTCM/12.                                             LASV346   003550
      A=ALPHA(ITRGT)                                                    LASV347   003560
      B=BETA(ITRGT)                                                     LASV348   003570
      NSID=NSIDE(ITRGT)                                                 LASV349   003580
      ARAD=A*DGTOR                                                      LASV350   003590
      IF (B.NE.0.) ARAD=B*DGTOR                                         LASV351   003600
      SINA=SIN(ARAD)                                                    LASV352   003610
      COSA=COS(ARAD)                                                    LASV353   003620
      IF (ABS(ABS(ARAD)-HLFPI).GT..0001) TANA=TAN(ARAD)                 LASV354   003630
      H=(HOB-T)*2.*MTOFT/CELL                                           LASV355   003640
      HSINW=H*SINW                                                      LASV356   003650
      H2=H*H                                                            LASV357   003660
C                                                                       LASV358   003670
C     RESET FLAGS                                                       LASV359   003680
      IFLAG=0                                                           LASV360   003690
```

```
      VOID2=0                                                           LASV361   003700
      IF (NZ.EQ.0) CALL FLAT                                            LASV362   003710
      IF (A.NE.0..AND.NSID.EQ.2) CALL A2SID                             LASV363   003720
      IF (A.NE.0..AND.NSID.EQ.1) CALL A1SID                             LASV364   003730
      IF (B.NE.0..AND.NSID.EQ.2) CALL B2SID                             LASV365   003740
      IF (B.NE.0..AND.NSID.EQ.1) CALL B1SID                             LASV366   003750
      SCALE2=SCALE1(ITRGT)                                              LASV367   003760
      IF (ABS(SCALE2).LT..00001) SCALE2=1.                              LASV368   003770
      IF (ABS(SCALE2-1.).LT..00001) GO TO 330                           LASV369   003780
      SCALE3=1./SCALE2                                                  LASV370   003790
      DO 320 I=LYMXL,LYMXR                                              LASV371   003800
      DO 320 J=LYMYB,LYMYT                                              LASV372   003810
  320 LAMDA2(I,J)=LAMDA2(I,J)*SCALE3                                    LASV373   003820
C                                                                       LASV374   003830
  330 BUFFER(1)=FILES                                                   LASV375   003840
      BUFFER(2)=NXX                                                     LASV376   003850
      BUFFER(3)=NYY                                                     LASV377   003860
      ABUFFR(4)=CELL                                                    LASV378   003870
      ABUFFR(5)=HOB                                                     LASV379   003880
      BUFFER(6)=IROW                                                    LASV380   003890
      BUFFER(7)=ICOL                                                    LASV381   003900
      BUFFER(8)=LYMYB                                                   LASV382   003910
      BUFFER(9)=LYMYT                                                   LASV383   003920
      BUFFER(10)=LYMXL                                                  LASV384   003930
      BUFFER(11)=LYMXR                                                  LASV385   003940
      ABUFFR(12)=A                                                      LASV386   003950
      ABUFFR(13)=B                                                      LASV387   003960
      BUFFER(14)=NSID                                                   LASV388   003970
      BUFFER(15)=NX                                                     LASV389   003980
      BUFFER(16)=NY                                                     LASV390   003990
      BUFFER(17)=NZ                                                     LASV391   004000
      ABUFFR(18)=TT(ITRGT)                                              LASV392   004010
      ABUFFR(19)=THCK                                                   LASV393   004020
      K=19                                                              LASV394   004030
      DO 340 I=LYMXL,LYMXR                                              LASV395   004040
      DO 340 J=LYMYB,LYMYT                                              LASV396   004050
      K=K+1                                                             LASV397   004060
  340 ABUFFR(K)=LAMDA2(I,J)                                             LASV398   004070
      CALL WRITMS (TAPE,BUFFER(1),K,FILES,-1,0)                         LASV399   004080
      PLIST(FILES,2)=K                                                  LASV400   004090
C     LOOP ON THE DIFFERENT CRITERIA                                    LASV401   004100
      DO 480 IKRIT=1,NKRIT                                              LASV402   004110
      WRITE (6,1150) KRIT(IKRIT),HOB,THCK,CELL1,NX,NY,NZ,TT(ITRGT),     LASV403W  004120
     1  SCALE2                                                          LASV404W  004130
      LINES=6                                                           LASV405   004140
      IF (NZ.EQ.0) GO TO 350                                            LASV406   004150
      WRITE (6,1160) A,B,NSID                                           LASV407W  004160
      LINES=LINES+2                                                     LASV408   004170
  350 WRITE (6,1170) NXX,NYY,SHELX,SHELY                                LASV409W  004180
      LINES=LINES+2                                                     LASV410   004190
      IF (IFLAG-8.LT.0) GO TO 360                                       LASV411   004200
      WRITE (6,1190) WORD(4),WORD(2),LYMYT                              LASV412W  004210
      LINES=LINES+2                                                     LASV413   004220
      IFLAG=IFLAG-8                                                     LASV414   004230
      GO TO 370                                                         LASV415   004240
  360 IF (LIMYT.NE.NYY) GO TO 370                                       LASV416   004250
      WRITE (6,1190) WORD(3),WORD(2),LYMYT                              LASV417W  004260
      LINES=LINES+2                                                     LASV418   004270
  370 IF (IFLAG-4.LT.0) GO TO 380                                       LASV419   004280
      WRITE (6,1190) WORD(4),WORD(2),LYMYB                              LASV420W  004290
```

```
      LINES=LINES+2                                                     LASV421  004300
      IFLAG=IFLAG-4                                                     LASV422  004310
      GO TO 390                                                         LASV423  004320
  380 IF (LIMYB.NE.1.OR.IROW.EQ.0) GO TO 390                            LASV424  004330
      WRITE (6,1190) WORD(3),WORD(2),LYMYB                              LASV425W 004340
      LINES=LINES+2                                                     LASV426  004350
  390 IF (IFLAG-2.LT.0) GO TO 400                                       LASV427  004360
      WRITE (6,1190) WORD(4),WORD(1),LYMXR                              LASV428W 004370
      LINES=LINES+2                                                     LASV429  004380
      IFLAG=IFLAG-2                                                     LASV430  004390
      GO TO 410                                                         LASV431  004400
  400 IF (LIMXR.NE.NXX) GO TO 410                                       LASV432  004410
      WRITE (6,1190) WORD(3),WORD(1),LYMXR                              LASV433W 004420
      LINES=LINES+2                                                     LASV434  004430
  410 IF (IFLAG.NE.1) GO TO 420                                         LASV435  004440
      WRITE (6,1190) WORD(4),WORD(1),LYMXL                              LASV436W 004450
      LINES=LINES+2                                                     LASV437  004460
      GO TO 430                                                         LASV438  004470
  420 IF (LIMXL.NE.1) GO TO 430                                         LASV439  004480
      WRITE (6,1190) WORD(3),WORD(1),LYMXL                              LASV440W 004490
      LINES=LINES+2                                                     LASV441  004500
  430 WRITE (6,1200) TNAME,FILES                                        LASV442W 004510
      LINES=LINES+2                                                     LASV443  004520
C                                                                       LASV444  004530
      LAP1=0                                                            LASV445  004540
      IF (VOID2.NE.0) GO TO 440                                         LASV446  004550
C     GO TO 440 IF PERF MATRIX IS ZEROES                                LASV447  004560
      CALL PEKAY (LAP1)                                                 LASV448  004570
  440 CALL OUTPUT (LAP1)                                                LASV449  004580
      IF (VOID2.EQ.0) GO TO 450                                         LASV450  004590
      WRITE (6,1230)                                                    LASV451W 004600
      LINES=LINES+2                                                     LASV452  004610
      GO TO 490                                                         LASV453  004620
  450 IF (CCC.EQ.10HCONTOUR   .AND.LAP1.NE.0) CALL CONTR                LASV454  004630
      IF (IKRIT.EQ.NKRIT) GO TO 470                                     LASV455  004640
C     RECOVER THE PERFORATION MATRIX                                    LASV456  004650
      K=19                                                              LASV457  004660
      DO 460 I=LYMXL,LYMXR                                              LASV456  004670
      DO 460 J=LYMYB,LYMYT                                              LASV459  004680
      K=K+1                                                             LASV460  004690
  460 LAMDA2(I,J)=ABUFFR(K)                                             LASV461  004700
  470 IF (LAP1.NE.0) GO TO 480                                          LASV462  004710
      IF (IKRIT.EQ.NKRIT) GO TO 480                                     LASV463  004720
      LAP1=NKRIT-IKRIT                                                  LASV464  004730
      WRITE (6,1220) FILES,TNAME,LAP1                                   LASV465W 004740
      LINES=31                                                          LASV466  004750
      GO TO 490                                                         LASV467  004760
  480 CONTINUE                                                          LASV468  004770
C                                                                       LASV469  004780
  490 IF (SHOW(ITRGT).EQ.0.OR.VOID2.NE.0) GO TO 510                     LASV470  004790
      WRITE (6,IFMT1) SHOW(ITRGT),HOB,CELL1                             LASV471W 004800
      WRITE (6,1210) TNAME,FILES                                        LASV472W 004810
      LINES=4                                                           LASV473  004820
C     RECOVER THE PERFORATION MATRIX                                    LASV474  004830
      K=19                                                              LASV475  004840
      DO 500 I=LYMXL,LYMXR                                              LASV476  004850
      DO 500 J=LYMYB,LYMYT                                              LASV477  004860
      K=K+1                                                             LASV478  004870
  500 LAMDA2(I,J)=ABUFFR(K)                                             LASV479  004880
      CALL SHOWW (ITRGT)                                                LASV480  004890
```

```
  510 FILES=FILES+1                                                           LASV481   004900
C                                                                             LASV482   004910
C                                                                             LASV483   004920
      IF (AAA(1).EQ.10HCONTOUR    .OR.AAA(1).EQ.10HNO PROBABI.OR.AAA(1).ELASV484R 004930
     1Q.10HFRAGMENTAT) READ (5,1090) (AAA(I),I=1,8)                           LASV485R  004940
      IF (AAA(1).NE.10HASSEMBLE  .AND.AAA(1).NE.10H ASSEMBLE .AND.AAA(1)LASV486   004950
     1.NE.10H  ASSEMBLE) GO TO 80                                             LASV487   004960
C                                                                             LASV488   004970
C                                                                             LASV489   004980
C     THIS PART OF MAIN ASSEMBLES A COMPLETE TARGET FROM ITS COMPONENT        LASV490   004990
C     PARTS                                                                   LASV491   005000
C                                                                             LASV492   005010
C                                                                             LASV493   005020
  520 KKSYM=1                                                                 LASV494   005030
      VOID2=0                                                                 LASV495   005040
  530 DECODE (50,1240,AAA(2) )FKSYM,FNPNTS,CCC,FNSTAP,NOPK                    LASV496*  005050
      KSYM=FKSYM+.1                                                           LASV497   005060
      NPNTS=FNPNTS+.1                                                         LASV498   005070
      NSTAPE=FNSTAP+.1                                                        LASV499   005080
      IF (NSTAPE.GT.0) GO TO 560                                              LASV500   005090
      DO 540 I=1,5000                                                         LASV501   005100
      J=5001-I                                                                LASV502   005110
      IF (SLIST(J,2).NE.0) GO TO 550                                          LASV503   005120
  540 CONTINUE                                                                LASV504   005130
      J=0                                                                     LASV505   005140
  550 NSTAPE=J+1                                                              LASV506   005150
  560 DO 570 I=1,7                                                            LASV507   005160
      SHFTX(I)=0                                                              LASV508   005170
  570 SHFTY(I)=0                                                              LASV509   005180
      IF (NPNTS.EQ.0) NPNTS=1                                                 LASV510   005190
      READ (5,1090) (AAA(I),I=1,8)                                            LASV511R  005200
      SHO=AAA(1)                                                              LASV512   005210
      IF (SHO.NE.10HEXPECTED P) GO TO 620                                     LASV513   005220
      DECODE (10,1100,AAA(3) )AAA(1),FSHOW(1)                                 LASV514*  005230
      IF (AAA(1).NE.10HALL=       ) GO TO 590                                 LASV515   005240
      DO 580 I=1,NPNTS                                                        LASV516   005250
  580 FSHOW(I)=FSHOW(1)                                                       LASV517   005260
      GO TO 600                                                               LASV518   005270
  590 READ (5,1110) (FSHOW(I),I=1,NPNTS)                                      LASV519R  005280
  600 DO 610 I=1,NPNTS                                                        LASV520   005290
  610 SHOW(I)=FSHOW(I)+.1                                                     LASV521   005300
      READ (5,1090) (AAA(I),I=1,8)                                            LASV522R  005310
  620 DECODE (80,1110,AAA(1) )FNKRIT,(FKRIT(I),I=1,7)                         LASV523*  005320
      NKRIT=FNKRIT+.1                                                         LASV524   005330
      DO 630 I=1,NKRIT                                                        LASV525   005340
  630 KRIT(I)=FKRIT(I)+.1                                                     LASV526   005350
      REWIND PTAPE1                                                           LASV527T  005360
      READ (5,1090) (AAA(I),I=1,8)                                            LASV528R  005370
      DO 950 IPNT=1,NPNTS                                                     LASV529   005380
  640 DECODE (80,1110,AAA(1) )FNFILE,(FFILE(I),I=1,7)                         LASV530*  005390
      NFILE=FNFILE+.1                                                         LASV531   005400
      DO 650 I=1,NFILE                                                        LASV532   005410
  650 FILE(I)=FFILE(I)+.1                                                     LASV533   005420
      READ (5,1090) (AAA(I),I=1,8)                                            LASV534R  005430
  660 IF (AAA(1).EQ.10HSHIFT X = .OR.AAA(1).EQ.10H SHIFT X =) GO TO 690       LASV535   005440
  670 IF (AAA(1).NE.10HSHIFT Y = .AND.AAA(1).NE.10H SHIFT Y =) GO TO 710LASV536   005450
      DECODE (60,1110,AAA(3) )(FSHFTY(I),I=2,NFILE)                           LASV537*  005460
      DO 680 I=2,NFILE                                                        LASV538   005470
  680 SHFTY(I)=FSHFTY(I)+SIGN(.1,FSHFTY(I))                                   LASV539   005480
      READ (5,1090) (AAA(I),I=1,8)                                            LASV540R  005490
```

```
      GO TO 660                                                         LASV541   005500
  690 DECODE (60,1110,AAA(3) )(FSHFTX(I),I=2,NFILE)                     LASV542*  005510
      DO 700 I=2,NFILE                                                  LASV543   005520
  700 SHFTX(I)=FSHFTX(I)+SIGN(.1,FSHFTX(I))                             LASV544   005530
      READ (5,1090) (AAA(I),I=1,8)                                      LASV545R  005540
      GO TO 670                                                         LASV546   005550
  710 ASSIGN 880 TO SNUM1                                               LASV547   005560
      CALL READMS (TAPE,BUFFER(1),PLIST(FILE(1),2),FILE(1))             LASV548   005570
      NXX=BUFFER(2)                                                     LASV549   005580
      NYY=BUFFER(3)                                                     LASV550   005590
      CELL=ABUFFR(4)                                                    LASV551   005600
      HOB=ABUFFR(5)                                                     LASV552   005610
      IROW1=BUFFER(6)                                                   LASV553   005620
      ICOL1=BUFFER(7)                                                   LASV554   005630
      L1MYB=BUFFER(8)                                                   LASV555   005640
      L1MYT=BUFFER(9)                                                   LASV556   005650
      L1MXL=BUFFER(10)                                                  LASV557   005660
      L1MXR=BUFFER(11)                                                  LASV558   005670
      B1=ABUFFR(13)                                                     LASV559   005680
      NSID1=BUFFER(14)                                                  LASV560   005690
      K=19                                                              LASV561   005700
      DO 720 I1=L1MXL,L1MXR                                             LASV562   005710
      DO 720 J1=L1MYB,L1MYT                                             LASV563   005720
      K=K+1                                                             LASV564   005730
  720 PERF1(I1,J1)=ABUFFR(K)                                            LASV565   005740
C                                                                       LASV566   005750
      NGSLP=IROW1                                                       LASV567   005760
      IF (B1.NE.0..AND.NSID1.EQ.1) ASSIGN 890 TO SNUM1                  LASV568   005770
      KK1=2*IROW1+1                                                     LASV569   005780
      LYMYB=L1MYB                                                       LASV570   005790
      LYMYT=L1MYT                                                       LASV571   005800
      LYMXL=L1MXL                                                       LASV572   005810
      LYMXR=L1MXR                                                       LASV573   005820
      IF (NFILE.GT.1) GO TO 740                                         LASV574   005830
      DO 730 I=LYMXL,LYMXR                                              LASV575   005840
      DO 730 J=LYMYB,LYMYT                                              LASV576   005850
  730 LAMDA(I,J)=PERF1(I,J)                                             LASV577   005860
      GO TO 920                                                         LASV578   005870
  740 DO 910 IFILE=2,NFILE                                              LASV579   005880
      ASSIGN 850 TO SNUM2                                               LASV580   005890
      CALL READMS (TAPE,BUFFER(1),PLIST(FILE(IFILE),2),FILE(IFILE))     LASV581   005900
      NXX=BUFFER(2)                                                     LASV582   005910
      NYY=BUFFER(3)                                                     LASV583   005920
      CELL=ABUFFR(4)                                                    LASV584   005930
      HOB=ABUFFR(5)                                                     LASV585   005940
      IROW2=BUFFER(6)                                                   LASV586   005950
      ICOL2=BUFFER(7)                                                   LASV587   005960
      L2MYB=BUFFER(8)                                                   LASV588   005970
      L2MYT=BUFFER(9)                                                   LASV589   005980
      L2MXL=BUFFER(10)                                                  LASV590   005990
      L2MXR=BUFFER(11)                                                  LASV591   006000
      B2=ABUFFR(13)                                                     LASV592   006010
      NSID2=BUFFER(14)                                                  LASV593   006020
      K=19                                                              LASV594   006030
      DO 750 I=L2MXL,L2MXR                                              LASV595   006040
      DO 750 J=L2MYB,L2MYT                                              LASV596   006050
      K=K+1                                                             LASV597   006060
  750 PERF2(I,J)=ABUFFR(K)                                              LASV598   006070
C                                                                       LASV599   006080
      IF (B2.NE.0..AND.NSID2.EQ.1) ASSIGN 860 TO SNUM2                  LASV600   006090
```

```
      KK2=2*IROW2+1
C     SHFT IS THE AMOUNT BY WHICH THE GEOMETRICAL CENTER OF THIS PART OF
C     THE TARGET IS OFFSET FROM THE POINT DEFINED AS THE TARGET CENTER,
C     WHICH POINT IS ALWAYS IDENTICAL TO THE GEOMETRICAL CENTER OF THE
C     PART OF THE TARGET DEFINED BY FILE(1).
      DELTX=ICOL2-ICOL1+SHFTX(IFILE)
      DELTY=IROW2-IROW1+SHFTY(IFILE)
      SLIPX=0
      SLIPY=0
      IF (L2MXL-DELTX.LT.LYMXL) LYMXL=L2MXL-DELTX
      IF (L2MXR-DELTX.GT.LYMXR) LYMXR=L2MXR-DELTX
      IF (L2MYB-DELTY.LT.LYMYB) LYMYB=L2MYB-DELTY
      IF (L2MYT-DELTY.GT.LYMYT) LYMYT=L2MYT-DELTY
      IF (LYMXL.GE.1) GO TO 770
      IF (LYMXR.LT.DIMX) GO TO 760
      LYMXL=1
      GO TO 790
  760 SLIPX=DIMX-LYMXR
      IF (1-LYMXL.LT.SLIPX) SLIPX=1-LYMXL
      GO TO 790
  770 IF (LYMXR.LE.DIMX) GO TO 790
      IF (LYMXL.GT.1) GO TO 780
      LYMXR=DIMX
      GO TO 790
  780 SLIPX=1-LYMXL
      IF (DIMX-LYMXR.GT.SLIPX) SLIPX=DIMX-LYMXR
  790 IF (LYMYB.GE.1) GO TO 810
      IF (LYMYT.LT.DIMY.AND.NOSLP.NE.0) GO TO 800
      LYMYB=1
      GO TO 830
  800 SLIPY=DIMY-LYMYT
      IF (1-LYMYB.LT.SLIPY) SLIPY=1-LYMYB
      GO TO 830
  810 IF (LYMYT.LE.DIMY) GO TO 830
      IF (LYMYB.GT.1.AND.NOSLP.NE.0) GO TO 820
      LYMYT=DIMY
      GO TO 830
  820 SLIPY=1-LYMYB
      IF (DIMY-LYMYT.GT.SLIPY) SLIPY=DIMY-LYMYT
  830 LYMXL=LYMXL+SLIPX
      LYMXR=LYMXR+SLIPX
      LYMYB=LYMYB+SLIPY
      LYMYT=LYMYT+SLIPY
      IF (LYMXL.LT.1) LYMXL=1
      IF (LYMXR.GT.DIMX) LYMXR=DIMX
      IF (LYMYB.LT.1) LYMYB=1
      IF (LYMYT.GT.DIMY) LYMYT=DIMY
      DO 890 I=LYMXL,LYMXR
      I1=I-SLIPX
      I2=I1+DELTX
      DO 890 J=LYMYB,LYMYT
      J1=J-SLIPY
      J2=J1+DELTY
      TERM1=0.
      TERM2=0.
      IF (I2.LT.L2MXL.OR.I2.GT.L2MXR) GO TO 860
      IF (J2.LT.L2MYB.OR.J2.GT.L2MYT) GO TO 840
      TERM2=PERF2(I2,J2)
      GO TO 860
  840 GO TO SNUM2, (850,860)
```

```
850 K2=KK2-J2                                                             LASV661  006700
    IF (J2.LT.1.AND.K2.LE.L2MYT.AND.K2.GE.L2MYB) TERM2=PERF2(12,K2)       LASV662  006710
860 IF (I1.LT.L1MXL.OR.I1.GT.L1MXR) GO TO 890                             LASV663  006720
    IF (J1.LT.L1MYB.OR.J1.GT.L1MYT) GO TO 870                             LASV664  006730
    TERM1=PERF1(I1,J1)                                                    LASV665  006740
    GO TO 890                                                             LASV666  006750
870 GO TO SNUM1, (860,890)                                                LASV667  006760
880 K1=KK1-J1                                                             LASV668  006770
    IF (J1.LT.1.AND.K1.LE.L1MYT.AND.K1.GE.L1MYB) TERM1=PERF1(I1,K1)       LASV669  006780
890 LAMDA(I,J)=TERM1+TERM2                                                LASV670  006790
    IROW1=IROW1+SLIPY                                                     LASV671  006800
    ICOL1=ICOL1+SLIPX                                                     LASV672  006810
    IF (IFILE.EQ.NFILE) GO TO 920                                         LASV673  006820
    DO 900 I=LYMXL,LYMXR                                                  LASV674  006830
    DO 900 J=LYMYB,LYMYT                                                  LASV675  006840
900 PERF1(I,J)=LAMDA(I,J)                                                 LASV676  006850
    L1MXL=LYMXL                                                           LASV677  006860
    L1MXR=LYMXR                                                           LASV678  006870
    L1MYB=LYMYB                                                           LASV679  006880
    L1MYT=LYMYT                                                           LASV680  006890
    KK1=2*IROW1+1                                                         LASV681  006900
    IF (B2.NE.0..AND.NSID2.EQ.1.OR.SHFTY(IFILE).NE.0) ASSIGN 690 TO SNLASV682  006910
   1UM1                                                                   LASV683  006920
910 CONTINUE                                                              LASV684  006930
920 IF (KSYM.NE.2) GO TO 940                                              LASV685  006940
    IF (KKSYM.EQ.2) GO TO 930                                             LASV686  006950
    WRITE (PTAPE1) ICOL1,IROW1,LYMXL,LYMXR,LYMYB,LYMYT,NFILE,((LAMDA(ILASV687W 006960
   1 ,J),I=LYMXL,LYMXR),J=LYMYB,LYMYT),(FILE(I),SHFTX(I),SHFTY(I),I=1LASV688W 006970
   2 ,NFILE)                                                              LASV689W 006980
    BACKSPACE PTAPE1                                                      LASV690T 006990
    KKSYM=KKSYM+1                                                         LASV691  007000
    ICOL=ICOL1                                                            LASV692  007010
    IROW=IROW1                                                            LASV693  007020
    LOMXL=LYMXL                                                           LASV694  007030
    LOMXR=LYMXR                                                           LASV695  007040
    GO TO 640                                                             LASV696  007050
930 CALL PAIR                                                             LASV697  007060
    GO TO 950                                                             LASV698  007070
940 WRITE (PTAPE1) HOB,CELL,KSYM,ICOL1,IROW1,LYMXL,LYMXR,LYMYB,LYMYT,     LASV699W 007080
   1  NFILE,((LAMDA(I,J),I=LYMXL,LYMXR),J=LYMYB,LYMYT),(FILE(I),          LASV700W 007090
   2  SHFTX(I),SHFTY(I),I=1,NFILE)                                        LASV701W 007100
950 KKSYM=1                                                               LASV702  007110
C                                                                         LASV703  007120
    ENCODE (10,1250,IFMT1(13) )                                           LASV704* 007130
    IF (SHD.NE.10HEXPECTED P) GO TO 1050                                  LASV705  007140
    REWIND PTAPE1                                                         LASV706T 007150
    ENCODE (70,1280,IFMT1(1) )                                            LASV707* 007160
    IF (NPNTS.GT.1) ENCODE (20,1260,IFMT1(13) )                           LASV708* 007170
    ASSIGN 1020 TO JUMPS                                                  LASV709  007180
    IPNT=0                                                                LASV710  007190
960 IPNT=IPNT+1                                                           LASV711  007200
    ENCODE (10,1270,IFMT1(15) )IPNT                                       LASV712* 007210
    READ (PTAPE1) HOB,CELL,KSYM,ICOL1,IROW1,LYMXL,LYMXR,LYMYB,LYMYT,      LASV713R 007220
   1  NFILE,((LAMDA2(I,J),I=LYMXL,LYMXR),J=LYMYB,LYMYT),(FILE(I),         LASV714R 007230
   2  SHFTX(I),SHFTY(I),I=1,NFILE)                                        LASV715R 007240
    IF (SHOW(IPNT).EQ.0) GO TO 1030                                       LASV716  007250
    SHELX=FLOAT(ICOL1)+.5                                                 LASV717  007260
    SHELY=FLOAT(IROW1)+.5                                                 LASV718  007270
    CELL1=CELL*12.                                                        LASV719  007280
    WRITE (6,IFMT1) SHOW(IPNT),HOB,CELL1                                  LASV720W 007290
```

```
      LINES=1                                                           LASV721  007300
  970 ASSIGN 1000 TO JUMP4                                              LASV722  007310
  980 IF (NFILE.GT.1) GO TO 990                                         LASV723  007320
      WRITE (6,1300) FILE(1)                                            LASV724W 007330
      LINES=LINES+3                                                     LASV725  007340
      GO TO JUMP4, (1000,1010)                                          LASV726  007350
  990 KEY1=NFILE-1                                                      LASV727  007360
      ENCODE (10,1310,IFMT2(7) )KEY1                                    LASV728* 007370
      ENCODE (10,1320,IFMT2(13) )KEY1                                   LASV729* 007380
      WRITE (6,IFMT2) (FILE(I),I=1,NFILE),WORD(5),(SHFTX(I),I=1,NFILE), LASV730W 007390
     1  WORD(6),(SHFTY(I),I=1,NFILE)                                    LASV731W 007400
      LINES=LINES+7                                                     LASV732  007410
      GO TO JUMP4, (1000,1010)                                          LASV733  007420
 1000 WRITE (6,1170) NXX,NYY,SHELX,SHELY                                LASV734W 007430
      LINES=LINES+2                                                     LASV735  007440
      IF (KSYM.NE.2) GO TO JUMP5, (1020,1080)                           LASV736  007450
      READ (PTAPE1) NFILE,(FILE(I),SHFTX(I),SHFTY(I),I=1,NFILE)         LASV737R 007460
      ASSIGN 1010 TO JUMP4                                              LASV738  007470
      WRITE (6,1330)                                                    LASV739W 007480
      LINES=LINES+1                                                     LASV740  007490
      GO TO 980                                                         LASV741  007500
 1010 WRITE (6,1180) SHELX,SHELY                                        LASV742W 007510
      LINES=LINES+2                                                     LASV743  007520
      GO TO JUMP5, (1020,1080)                                          LASV744  007530
 1020 CALL SHOWW (IPNT)                                                 LASV745  007540
      GO TO 1040                                                        LASV746  007550
 1030 IF (KSYM.EQ.2) READ (PTAPE1)                                      LASV747R 007560
 1040 IF (IPNT.LT.NPNTS) GO TO 960                                      LASV748  007570
C                                                                       LASV749  007580
 1050 ENCODE (70,1290,IFMT1(1) )                                        LASV750* 007590
      ASSIGN 1080 TO JUMP5                                              LASV751  007600
      IKRIT=0                                                           LASV752  007610
 1060 IKRIT=IKRIT+1                                                     LASV753  007620
      REWIND PTAPE1                                                     LASV754T 007630
      IPNT=0                                                            LASV755  007640
 1070 IPNT=IPNT+1                                                       LASV756  007650
      READ (PTAPE1) HOB,CELL,KSYM,ICOL1,IROW1,LYMXL,LYMXR,LYMYB,LYMYT,  LASV757R 007660
     1  NFILE,((LAMDA2(I,J),I=LYMXL,LYMXR),J=LYMYB,LYMYT),(FILE(I),     LASV758R 007670
     2  SHFTX(I),SHFTY(I),I=1,NFILE)                                    LASV759R 007680
      ENCODE (10,1270,IFMT1(15) )IPNT                                   LASV760* 007690
      HHHOB(IPNT)=HOB                                                   LASV761  007700
      SHELX=FLOAT(ICOL1)+.5                                             LASV762  007710
      SHELY=FLOAT(IROW1)+.5                                             LASV763  007720
      CELL1=CELL*12.                                                    LASV764  007730
      WRITE (6,IFMT1) KRIT(IKRIT),HOB,CELL1                             LASV765W 007740
      LINES=1                                                           LASV766  007750
      GO TO 970                                                         LASV767  007760
 1080 CALL PEKAY (LAP1)                                                 LASV768  007770
C     FOLLOWING INSTRUCTION MERELY GUARANTEES LAP1 .NE. 0               LASV769  007780
      LAP1=NXX                                                          LASV770  007790
      CALL OUTPUT (LAP1)                                                LASV771  007800
      IF (CCC.EQ.10HCONTOUR    ) CALL CONTR                             LASV772  007810
      IF (IPNT.LT.NPNTS) GO TO 1070                                     LASV773  007820
      IF (IKRIT.LT.NKRIT) GO TO 1060                                    LASV774  007830
C                                                                       LASV775  007840
C                                                                       LASV776  007850
      IF (AAA(1).EQ.10HASSEMBLE  .OR.AAA(1).EQ.10H ASSEMBLE .OR.AAA(1).ELASV777  007860
     1Q.10H  ASSEMBLE) GO TO 530                                        LASV778  007870
      GO TO 80                                                          LASV779  007880
C                                                                       LASV780  007890
```

```
C                                                                       LASV781
 1090 FORMAT (8A10)                                                     LASV782
 1100 FORMAT (2X,A4,F4.1)                                               LASV783
 1110 FORMAT (8F10.4)                                                   LASV784
 1120 FORMAT (A10,7E10.4)                                               LASV785
 1130 FORMAT (7F10.4,2F5.3)                                             LASV786
 1140 FORMAT (1H1/20H THOR CONSTANTS FOR ,A10,7H    CV=,F7.3,4X,7HALPHAVLASV787
     1=,F7.3,4X,6HBETAV=,F7.3,4X,7HGAMMAV=,F7.3,4X,8HLAMBDAV=,F7.3,4X,3HLASV788
     2KV=,F7.3//1H ,33X,3HCM=,F7.3,4X,7HALPHAM=,F7.3,4X,6HBETAM=,F7.3,4XLASV789
     3,7HGAMMAM=,F7.3,4X,8HLAMBDAM=,F7.3,4X,3HKM=,F7.3/////26HOPROJECTILLASV790
     4E ANGLE OF FALL=,F5.1,8H DEGREES,6X,29HPROJECTILE TERMINAL VELOCITLASV791
     5Y=,F8.2,12H FEET/SECOND,6X,13HBLAST RADIUS=,F6.2,7H INCHES//14H SHLASV792
     6APE FACTOR=,F6.1,6X,2HA=,F5.3/)                                   LASV793
 1150 FORMAT (1H1/35H PROBABILITY OF SUSTAINING AT LEAST,I3,13H PERFORATLASV794
     1IONS//5H HOB=,F5.2,7H METERS,4X,10HTHICKNESS=,F4.2,7H INCHES,4X,5HLASV795
     2CELL=,F5.2,7H INCHES,4X,3HNX=,I3,6H CELLS,4X,3HNY=,I3,6H CELLS,4X,LASV796
     33HNZ=,I3,6H CELLS,4X,2HT=,F6.1,7H INCHES//42H SCALE FACTOR APPLIEDLASV797
     4 TO INPUT PLATE = 1./,F5.3/)                                      LASV798
 1160 FORMAT (7H ALPHA=,F6.1,8H DEGREES,4X,5HBETA=,F6.1,8H DEGREES,4X,I1LASV799
     1,6H SIDES/)                                                       LASV800
 1170 FORMAT (5H NXX=,I3,6H CELLS,4X,4HNYY=,I3,6H CELLS,4X,25HPROJECTILELASV801
     1 LOCATION WAS (,F6.1,1H,,F6.1,1H)/)                               LASV802
 1180 FORMAT (26H PROJECTILE LOCATION WAS (,F6.1,1H,,F6.1,1H)/)         LASV803
 1190 FORMAT (11H THE TARGET,A7,88HSUSTAIN ADDITIONAL PERFORATIONS WERE LASV804
     1ITS CENTER FURTHER REMOVED FROM THE PROJECTILE THAN,A8,I3/)       LASV805
 1200 FORMAT (86H THE EXPECTED PERFORATIONS MATRIX FOR THIS CASE IS ON TLASV806
     1HE EXPECTED PERFORATIONS FILE (,A10,18H) AS RECORD NUMBER,I5/)    LASV807
 1210 FORMAT (/73H THIS EXPECTED PERFORATIONS MATRIX IS ON THE EXPECTED LASV808
     1PERFORATIONS FILE (,A10,18H) AS RECORD NUMBER,I5/)                LASV809
 1220 FORMAT (1H1,27(/),1H ,28X,74HCOMPUTATION OF THE PROBABILITY OF DAMLASV810
     1AGE MATRICES FOR THE CURRENT EXPECTED/1H0,28X,34HPERFORATIONS MATRLASV811
     2IX (RECORD NUMBER,I5,28H, EXPECTED PERFORATIONS FILE,A10,9H) FOR TLASV812
     3HE,I3/1H0,28X,73HREMAINING CRITERIA HAS BEEN SUPPRESSED BECAUSE THLASV813
     4EY WOULD HAVE BEEN NULL.)                                         LASV814
 1230 FORMAT (/93H THEREFORE, COMPUTATION OF THE PROBABILITY OF DAMAGE MLASV815
     1ATRICES FOR ALL CRITERIA IS SUPPRESSED.)                          LASV816
 1240 FORMAT (2E10.4,A10,E10.4,A10)                                     LASV817
 1250 FORMAT (10HES)          )                                         LASV818
 1260 FORMAT (20HES,5X,15HPOINT NUMBE)                                  LASV819
 1270 FORMAT (1HR,I3,6H)      )                                         LASV820
 1280 FORMAT (70H(1H1/  28H EXPECTED PERFORATIONS*10.**,I2,11H, TRUNCATELASV821
     1D,             )                                                  LASV822
 1290 FORMAT (70H(1H1/  35H PROBABILITY OF SUSTAINING AT LEAST,I3,13H PELASV823
     1RFORATIONS,     )                                                 LASV824
 1300 FORMAT (/52H THIS TARGET/HOB COMBINATION WAS OBTAINED FROM RECORD,LASV825
     1I5/)                                                              LASV826
 1310 FORMAT (I2,8H(I5,1H,))                                            LASV827
 1320 FORMAT (5HWERE,,I2,3H(I5)                                         LASV828
 1330 FORMAT (130H LOW ORDER ROWS WERE OBTAINED BY REFLECTING THE TARGETLASV829
     1/HOB COMBINATION WITH OPPOSITE ORIENTATION THROUGH THE PROJECTILE LASV830
     2TRAJECTORY)                                                       LASV831
 1340 FORMAT (1H ,10(/),1H ,46X,39HTHE FRAGMENTATION DATA HAVE NOT CHANGLASV832
     1ED)                                                               LASV833
 1350 FORMAT (1H1,27(/),1H ,34X,63HA CARD WITH AN INCORRECT ID FIELD (COLASV834
     1LUMNS 1-10) HAS BEEN READ./1H0,34X,62HTHIS IS A FATAL ERROR. THE CLASV835
     2ARD IMAGE FOLLOWS BETWEEN THE '*'.//1H0,25X,1H*,8A10,1H*)         LASV836
 1360 FORMAT (1H1)                                                      LASV837
      END                                                               LASV838-
```

```
      BLOCK DATA                                                        * 838* 1 008480
      COMMON /DISC/ DISC(15016)                                         BLOKD 2  008490
      INTEGER DISC                                                      BLOKD 3  008500
      DATA DISC /15016*0/                                               BLOKD 4D 008510
      END                                                               BLOKD 5- 008520



      SUBROUTINE FILNAM (NAME3,NAME4)                                   * 843* 2 008530
C     THIS SUBROUTINE RETURNS THE NAMES OF THE 3RD AND 4TH FILES AS     FILNM 2  008540
C     SPECIFIED ON THE PROGRAM CARD OR ON A CONTROL CARD EQUIVALENCING  FILNM 3  008550
C     A FILE NAME TO THESE FILES.                                       FILNM 4  008560
C                                                                       FILNM 5  008570
      DIMENSION DUM(1)                                                  FILNM 6  008580
      I=LOCF(DUM(1))                                                    FILNM 7  008590
      I=-I                                                              FILNM 8  008600
      J=IBITS(DUM(I+5),0,18)                                            FILNM 9  008610
      K=J+I+1                                                           FILNM10  008620
      NAME3=IBITS(DUM(K),18,42)                                         FILNM11  008630
      J=IBITS(DUM(I+6),0,18)                                            FILNM12  008640
      K=J+I+1                                                           FILNM13  008650
      NAME4=IBITS(DUM(K),18,42)                                         FILNM14  008660
      NAME3=SHIFT(NAME3,18)                                             FILNM15  008670
      NAME4=SHIFT(NAME4,18)                                             FILNM16  008680
      CALL BLFILL (NAME3,NAME3)                                         FILNM17  008690
      CALL BLFILL (NAME4,NAME4)                                         FILNM18  008700
      RETURN                                                            FILNM19  008710
      END                                                               FILNM20- 008720



      INTEGER FUNCTION USED (DUMMY)                                     * 863* 3 008730
C     THIS FUNCTION RETURNS THE NUMBER OF LINES USED ON THE CURRENT PAGEUSED  2  008740
C                                                                       USED  3  008750
      COMMON /PAGE/ LINES, PAGE                                         USED  4  008760
      INTEGER PAGE                                                      USED  5  008770
      IF (LINES-63) 20,20,30                                            USED  6  008780
   10 LINES=LINES+60                                                    USED  7  008790
   20 USED=LINES                                                        USED  8  008800
      RETURN                                                            USED  9  008810
   30 LINES=LINES-(LINES/PAGE)*PAGE                                     USED 10  008820
      IF (LINES-3) 10,10,20                                             USED 11  008830
      END                                                               USED 12- 008840
```

```
      SUBROUTINE FRAG (KEY)                                             * 875* 4  008850
C                                                                       FRAG  2   008860
      COMMON /COM3/COM3(10000)                                          FRAG  3   008870
      DIMENSION SPRBS(46),VSTMN(46),VSTMX(46),NSP(46),FNSP(46),M(40,46) FRAG  4   008880
     1          ,Q(40,46),SGST(46)                                      FRAG  5   008890
      EQUIVALENCE  (COM3(1),SPRBS(1)),(COM3(139),VSTMN(1))              FRAG  6   008900
     1            ,(COM3(185),VSTMX(1)),(COM3(323),NSP(1),FNSP(1))      FRAG  7   008910
     2            ,(COM3(369),M(1,1)),(COM3(2209),Q(1,1))               FRAG  8   008920
     3            ,(COM3(4049),SGST(1)),(COM3(9936),NSPRA,FNSPRA)       FRAG  9   008930
     4            ,(COM3(9940),SHAPE),(COM3(9941),AA),(COM3(9942),NNSP) FRAG 10   008940
     5            ,(COM3(9983),K22),(COM3(9984),K222)                   FRAG 11   008950
C                                                                       FRAG 12   008960
      COMMON /AAA/AAA(8)                                                FRAG 13   008970
      COMMON /CONSTT/CONSTT(6)                                          FRAG 14   008980
      EQUIVALENCE (CONSTT(1),TWOPI),(CONSTT(2),RTODG),(CONSTT(3),DGTOR) FRAG 15   008990
      COMMON /NUM46/NUM(46)                                             FRAG 16   009000
C                                                                       FRAG 17   009010
      DIMENSION WORD(3)                                                 FRAG 18   009020
C                                                                       FRAG 19   009030
      INTEGER AAA,WORD                                                  FRAG 20   009040
      REAL M,NUM                                                        FRAG 21   009050
C                                                                       FRAG 22   009060
      DATA (WORD(I),I=1,3) /6H M    , 6H Q    , 6H      /               FRAG 23D  009070
C                                                                       FRAG 24   009080
      ENTRY FRAG1                                                       FRAG 25E  009090
      DECODE (20,190,AAA(3) )SHAPE,AA                                   FRAG 26*  009100
      READ (5,190) FNSPRA,(FNSP(I),I=1,7)                               FRAG 27R  009110
      NSPRA=FNSPRA+.1                                                   FRAG 28   009120
      IF (NSPRA.GT.7) READ (5,190) (FNSP(I),I=8,NSPRA)                  FRAG 29R  009130
      DO 10 I=1,NSPRA                                                   FRAG 30   009140
   10 NSP(I)=FNSP(I)+.1                                                 FRAG 31   009150
      DO 40 K2=1,NSPRA                                                  FRAG 32   009160
      READ (5,180) (Q(I,K2),M(I,K2),I=1,3),Q(4,K2)                      FRAG 33R  009170
      K1=NSP(K2)                                                        FRAG 34   009180
      IF (K1-4) 40,20,30                                                FRAG 35   009190
   20 READ (5,190) M(4,K2)                                              FRAG 36R  009200
      GO TO 40                                                          FRAG 37   009210
   30 READ (5,190) (M(I-1,K2),Q(I,K2),I=5,K1),M(K1,K2)                  FRAG 38R  009220
   40 CONTINUE                                                          FRAG 39   009230
      DO 50 I=1,NSPRA                                                   FRAG 40   009240
      READ (5,160) SPRBS(I),SPRBS(I+1),VSTMN(I),VSTMX(I),NUM(I),SGST(I) FRAG 41R  009250
   50 CONTINUE                                                          FRAG 42   009260
      CSBS2=1.                                                          FRAG 43   009270
      DO 60 I=1,NSPRA                                                   FRAG 44   009280
      SPRBS(I+1)=SPRBS(I+1)*DGTOR                                       FRAG 45   009290
C     NO NEED TO CONVERT SPRBS(1)=0.                                    FRAG 46   009300
      CSBS1=CSBS2                                                       FRAG 47   009310
      CSBS2=COS(SPRBS(I+1))                                             FRAG 48   009320
      SOLANG=TWOPI*(CSBS1-CSBS2)                                        FRAG 49   009330
      IF (NUM(I).LT..0000001) NUM(I)=SGST(I)*SOLANG                     FRAG 50   009340
      IF (SGST(I).LT..0000001) SGST(I)=NUM(I)/SOLANG                    FRAG 51   009350
   60 CONTINUE                                                          FRAG 52   009360
C     FIND THE BLOCKS OF DYNAMIC SPRAYS, FRONT AND REAR, FOR WHICH SGST FRAG 53   009370
C     IS ZERO TO ELIMINATE THEM FROM CONSIDERATION                      FRAG 54   009380
      DO 70 K22=1,NSPRA                                                 FRAG 55   009390
      IF (SGST(K22).GT..0000001) GO TO 80                               FRAG 56   009400
   70 CONTINUE                                                          FRAG 57   009410
   80 DO 90 K=1,NSPRA                                                   FRAG 58   009420
      K222=NSPRA-K+1                                                    FRAG 59   009430
      IF (SGST(K222).GT..0000001) GO TO 170                             FRAG 60   009440
```

```
 90 CONTINUE
    GO TO 170
C
    ENTRY FRAG2
    WRITE (6,210) KEY
    DO 140 K2=1,NSPRA
    WRITE (6,240) K2
    NNSP=NSP(K2)
    KNT1=NNSP/10
    IF (KNT1*10.EQ.NNSP) KNT1=KNT1-1
    III=0
    IIII=1
    IF (KNT1.LE.0) GO TO 110
    DO 100 I=1,KNT1
    II=(I-1)*10+1
    III=I*10
    WRITE (6,230) WORD(IIII),(M(K1,K2),K1=II,III)
100 IIII=3
110 III=III+1
    WRITE (6,230) WORD(IIII),(M(K1,K2),K1=III,NNSP)
    III=0
    IIII=2
    IF (KNT1.LE.0) GO TO 130
    DO 120 I=1,KNT1
    II=(I-1)*10+1
    III=I*10
    WRITE (6,230) WORD(IIII),(Q(K1,K2),K1=II,III)
120 IIII=3
130 III=III+1
    WRITE (6,230) WORD(IIII),(Q(K1,K2),K1=III,NNSP)
140 CONTINUE
    WRITE (6,200)
    DO 150 I=1,NSPRA
    SPRBS(I+1)=SPRBS(I+1)*RTODG
    WRITE (6,220) I,SPRBS(I),SPRBS(I+1),VSTMN(I),VSTMX(I),NUM(I),
   1  SGST(I)
150 CONTINUE
    DO 160 I=1,NSPRA
160 SPRBS(I+1)=SPRBS(I+1)*DGTOR
170 RETURN
C
180 FORMAT (10X,7E10.4)
190 FORMAT (8E10.4)
200 FORMAT (//1H0,6X,5HSPRAY,6X,7HLEADING,5X,8HTRAILING,6X,6HMEDIAN,6X
   1,8HMAXIMUM ,2(5X,8HFRAGMENT)/1H ,6X,6HNUMBER,4X,2(6H  EDGE,7X),2(8
   2HVELOCITY,5X),7H NUMBER,6X,7HDENSITY/)
210 FORMAT (A1/31H THE FRAGMENTATION DATA FOLLOW.//)
220 FORMAT (1H ,I10,6F13.1)
230 FORMAT (1H ,A6,10E12.4)
240 FORMAT (//13H SPRAY NUMBER,I3/)
    END
```

```
      SUBROUTINE VECTR
C     THIS SUBROUTINE VECTORS THE SPRAYS FORWARD TO ACCOUNT FOR THE
C     PROJECTILE TERMINAL VELOCITY.
C     THIS ROUTINE EXPECTS THE VELOCITIES TO CORRESPOND TO THE SPRAY
C     MIDPOINTS. THE 1ST AND LAST SPRAYS MUST BE CONES CENTERED AT 0
C     AND 180 DEGREES, RATHER THAN CONICAL ANNULI. NOTE THAT THE
C     MIDSPRAY POINT FOR THESE 2 SPRAYS IS AT THE OUTER SPRAY BOUNDARY,
C     RATHER THAN AT THE AVERAGE OF THE 2 BOUNDARIES.
C
      COMMON /COM3/COM3(10000)
      DIMENSION SPRBS(46),SPRBD(92),VST(92),VDYN(92),SGDIN(92)
      EQUIVALENCE  (COM3(1),SPRBS(1)),(COM3(47),SPRBD(1))
     1            ,(COM3(139),VST(1)),(COM3(231),VDYN(1))
     2            ,(COM3(4095),SGDIN(1)),(COM3(9938),NSPRA)
     3            ,(COM3(9961),VTERM)
C
      COMMON /CONSTT/CONSTT(6)
      EQUIVALENCE (CONSTT(1),TWOPI)
      COMMON /NUM46/NUM(46)
C
      REAL NUM
C
C     RATHER THAN OBTAIN THE ANGLES AT WHICH THE DYNAMIC SPRAY EDGES
C     OCCUR, WE LEAVE THEM AS COSINES
C
C     DO THE 1ST HALF OF THE 92 WORD ARRAYS-IE, V STATIC MEDIAN.
      II=NSPRA-1
      VTM2=VTERM**2
      K=0
C     LEADING EDGE OF FIRST SPRAY
   10 CSBS1=1.
      SPRBD(1+K)=1.
      DO 20 I=1,II
      J=I+K
C     POSITIONS OF DYNAMIC SPRAY EDGES
      CSBS2=COS(SPRBS(I+1))
      VBNDS=(VST(J)+VST(J+1))/2.
      VBNDD=SQRT(VBNDS**2+2.*VBNDS*VTERM*CSBS2+VTM2)
      SPRBD(J+1)=(VBNDS*CSBS2+VTERM)/VBNDD
C     VECTOR THE DENSITIES
      SGDIN(J)=NUM(I)/(ABS(SPRBD(J+1)-SPRBD(J))*TWOPI)
C     VECTOR THE STATIC SPRAY SPEEDS (CORRESPOND TO SPRAY MID POINT)
      SPRMS=(SPRBS(I)+SPRBS(I+1))/2.
      VDYN(J)=SQRT(VST(J)**2+2.*VST(J)*VTERM*COS(SPRMS)+VTM2)
C     SET UP FOR NEXT SPRAY.
   20 CSBS1=CSBS2
C     THE 1ST SPRAY
      VDYN(1+K)=VST(1+K)+VTERM
C     THE LAST SPRAY.
      SGDIN(NSPRA+K)=NUM(NSPRA)/(ABS(1.+SPRBD(NSPRA+K))*TWOPI)
      VDYN(NSPRA+K)=VST(NSPRA+K)-VTERM
      SPRBD(NSPRA+1+K)=-1.
C     FINISHED WITH VSTMN, DO VSTMX, OR, FINISHED WITH VSTMX, RETURN
      IF (K.EQ.46) RETURN
      K=46
      GO TO 10
      END
```

```
      SUBROUTINE GEOMT                                                  *1043* 6  010530
C     THIS SUBROUTINE DOES THE GEOMETRY AND CALLS DAMGE FOR THE TARGET  GEOMT 2   010540
C     THETA IS THE POLAR ANGLE BETWEEN THE PROJECTILE TRAJECTORY AND THEGEOMT 3   010550
C     POSITION VECTOR TO THE CHOSEN CELL. ETA IS RELATED TO THE         GEOMT 4   010560
C     ANGLE BETWEEN THE POSITION VECTOR AND THE CELL NORMAL. NOTE THAT  GEOMT 5   010570
C     WE DON'T ACTUALLY GET THE ANGLES BUT RATHER DEAL WITH THEIR TRIG  GEOMT 6   010580
C     FUNCTIONS. STER IS THE PRESENTED AREA OF THE CHOSEN CELL IN       GEOMT 7   010590
C     STERADIANS.                                                       GEOMT 8   010600
C                                                                       GEOMT 9   010610
      COMMON /COM1/COM1(20100)                                          GEOMT10   010620
      DIMENSION PERF(100,100)                                           GEOMT11   010630
      EQUIVALENCE (COM1(1),PERF(1,1))                                   GEOMT12   010640
C                                                                       GEOMT13   010650
      COMMON /COM3/COM3(10000)                                          GEOMT14   010660
      EQUIVALENCE  (COM3(9939),NNSPRA),(COM3(9960),EPSLN)               GEOMT15   010670
     1             ,(COM3(9965),NX),(COM3(9966),NY),(COM3(9967),NZ)     GEOMT16   010680
     2             ,(COM3(9969),A),(COM3(9970),B),(COM3(9979),LIMXL)    GEOMT17   010690
     3             ,(COM3(9980),LIMXR),(COM3(9981),LIMYB)               GEOMT18   010700
     4             ,(COM3(9982),LIMYT),(COM3(9985),H),(COM3(9986),H2)   GEOMT19   010710
     5             ,(COM3(9987),SINW),(COM3(9988),COSW)                 GEOMT20   010720
     6             ,(COM3(9989),HSINW),(COM3(9990),SINA,SINB)           GEOMT21   010730
     7             ,(COM3(9991),COSA,COSB),(COM3(9992),TANA,TANB)       GEOMT22   010740
     8             ,(COM3(9993),R,RA,RB),(COM3(9994),THETA)             GEOMT23   010750
     9             ,(COM3(9995),STER),(COM3(9996),PRF1),(COM3(9999),ETA) GEOMT24  010760
C                                                                       GEOMT25   010770
      COMMON /COLROW/ICOL,IROW,ICOL1,IROW1                              GEOMT26   010780
      COMMON /NXXNYY/NXX,NYY                                            GEOMT27   010790
      COMMON /VOID2/VOID2                                               GEOMT28   010800
C                                                                       GEOMT29   010810
      INTEGER VOID2                                                     GEOMT30   010820
C                                                                       GEOMT31   010830
      ENTRY FLAT                                                        GEOMT32E  010840
C     THIS ENTRY HANDLES A SINGLE HORIZONTAL PLATE                      GEOMT33   010850
      ETA=H                                                             GEOMT34   010860
C                                                                       GEOMT35   010870
C     LOOP ON THE CELLS                                                 GEOMT36   010880
      IF (H.GT.C.) GO TO 10                                             GEOMT37   010890
      VOID2=2                                                           GEOMT38   010900
      GO TO 40                                                          GEOMT39   010910
   10 IF (NNSPRA.GT.0) GO TO 20                                         GEOMT40   010920
      VOID2=1                                                           GEOMT41   010930
      GO TO 40                                                          GEOMT42   010940
   20 DO 30 JJ=1,NYY                                                    GEOMT43   010950
      FJJ=2*(JJ-IROW)-1                                                 GEOMT44   010960
      DO 30 II=1,NXX                                                    GEOMT45   010970
C     BUILD A CELL. DO EVERYTHING IN TERMS OF HALF CELLS. XX(II,JJ)=FII GEOMT46   010980
C     AND YY(II,JJ)=FJJ                                                 GEOMT47   010990
      FII=2*(II-ICOL)-1                                                 GEOMT48   011000
      R2=FII**2+FJJ**2+H2                                               GEOMT49   011010
      R=SQRT(R2)                                                        GEOMT50   011020
      THETA=(HSINW+FII*COSW)/R                                          GEOMT51   011030
      STER=4.*H/(R2*R)                                                  GEOMT52   011040
C     CELL GEOMETRY IS COMPLETE                                         GEOMT53   011050
C                                                                       GEOMT54   011060
      CALL DAMGE                                                        GEOMT55   011070
C                                                                       GEOMT56   011080
C     THE NEXT STATEMENT COMPLETES THIS CELL, THEN DO THE NEXT CELL     GEOMT57   011090
   30 PERF(II,JJ)=PRF1/2.                                               GEOMT58   011100
C                                                                       GEOMT59   011110
C     PROJECTILE IMPACT ON THE PLATE                                    GEOMT60   011120
```

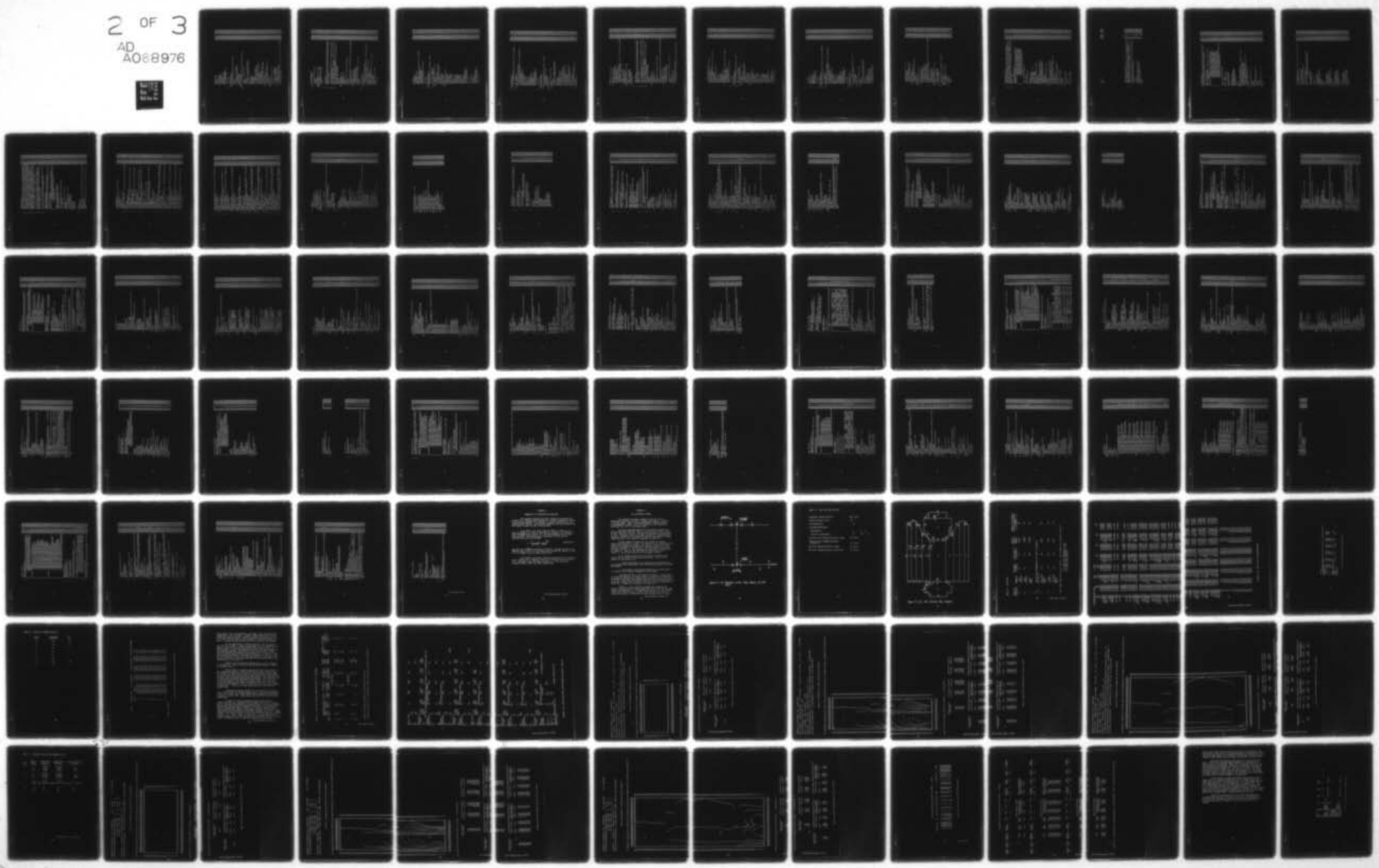
```
   40 IF (H.GT.EPSLN) GO TO 880
      F=EPSLN/2.
      IF (H.GT.0.) F=SQRT(EPSLN**2-H**2)/2.
      IF (AINT(F).LT.F) F=F+1.
      K=F
      LIMXL=ICOL-K
      LIMXR=ICOL+K
      LIMYB=IROW-K
      LIMYT=IROW+K
      IF (LIMXL.LT.1) LIMXL=1
      IF (LIMXR.GT.NXX) LIMXR=NXX
      IF (LIMYB.LT.1) LIMYB=1
      IF (LIMYT.GT.NYY) LIMYT=NYY
      DO 50 I=LIMXL,LIMXR
      DO 50 J=LIMYB,LIMYT
C     ASSUME 10000. IS LARGE ENOUGH TO YIELD PK=1. FOR ANY PLATE
C     COMBINATION AND ANY KILL CRITERION
   50 PERF(I,J)=10000.
      IF (VOID2.EQ.0) GO TO 890
      VOID2=0
      GO TO 900
C
C
      ENTRY A1SID
C     THIS ENTRY HANDLES A SINGLE SIDE WHERE THE JUNCTION OF THAT SIDE
C     WITH THE TOP IS PARALLEL TO THE Y AXIS. IN THIS ENTRY, ALPHA
C     IS POSITIVE (NEGATIVE) WHEN THE OUTWARD DRAWN NORMAL TO THE SIDE
C     HAS A COMPONENT IN THE POSITIVE (NEGATIVE) X DIRECTION.
C
      ICOL=ICOL+1
      IFSTA=1
      ILSTB=NXX
      COSAA=COSA
      SINAA=SINA
      IF (A.GT.0.) GO TO 60
      COSAA=-COSAA
      SINAA=-SINAA
   60 IF (NNSPRA.GT.0) GO TO 70
      VOID2=1
      GO TO 140
   70 VAL=0.
      IF (ABS(ABS(A)-90.).GT..0001) VAL=H/(2.*TANA)
      IVAL=VAL
      F=IVAL
      IF (F.NE.VAL.AND.(90.-ABS(A).GT..0001.AND.H.GE.0..OR.ABS(A)-90..GT
     1..0001.AND.H.LT.0.)) GO TO 80
      IVAL=IVAL-1
      IF (A.LT.0.) IVAL=IVAL+2
   80 IF (A.GT.0..AND.IVAL+ICOL.LT.NXX) ILSTB=IVAL+ICOL
      IF (A.LT.0..AND.IVAL+ICOL.GT.1) IFSTA=IVAL+ICOL
      IF (ILSTB.GE.1.AND.IFSTA.LE.NXX) GO TO 90
      VOID2=2
      GO TO 140
C
C     II SPECIFIES THE COLUMN BOUNDARY AT WHICH THE TOP EDGE OF THE SIDE
C     IS LOCATED. THE COLUMN BOUNDARY NUMBERED 'K' IS ON THE LOW ORDER
C     SIDE OF COLUMN K.
   90 IF (H.GE.0..OR.ABS(A)-90..LT..0001) GO TO 100
      IF (ILSTB.GT.ICOL.AND.A.GT.0.) ILSTB=ICOL
      IF (IFSTA.LT.ICOL.AND.A.LT.0.) IFSTA=ICOL
```

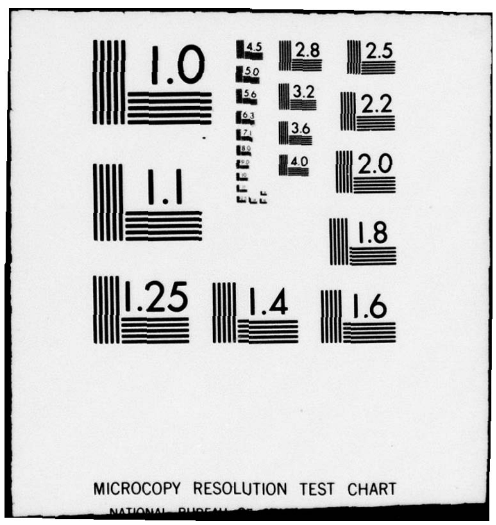
```
      IF (IFSTA.LE.ILSTB) GO TO 200                                     GEOM121   011730
      VOID2=2                                                           GEOM122   011740
      GO TO 210                                                         GEOM123   011750
  100 DO 120 II=IFSTA,ILSTB                                             GEOM124   011760
      XX=2*(II-ICOL)                                                    GEOM125   011770
      DO 120 JJ=1,NYY                                                   GEOM126   011780
      Y=2*(JJ-IROW)-1                                                   GEOM127   011790
      PRFK=0.                                                           GEOM128   011800
      DO 110 K=1,NZ                                                     GEOM129   011810
      FK=2*K-1                                                          GEOM130   011820
      X=XX+FK*COSAA                                                     GEOM131   011830
      Z=-H-FK*SINAA                                                     GEOM132   011840
      R2=X**2+Y**2+Z**2                                                 GEOM133   011850
      R=SQRT(R2)                                                        GEOM134   011860
      ETA=-Z*COSA-X*SINA                                                GEOM135   011870
      THETA=(X*COSW-Z*SINW)/R                                           GEOM136   011880
      STER=4.*ETA/(R2*R)                                                GEOM137   011890
C                                                                       GEOM138   011900
      CALL DAMGE                                                        GEOM139   011910
C                                                                       GEOM140   011920
  110 PRFK=PRFK+PRF1                                                    GEOM141   011930
  120 PERF(II,JJ)=PRFK/2.                                               GEOM142   011940
      IF (IFSTA.EQ.1.AND.ILSTB.EQ.NXX) GO TO 140                        GEOM143   011950
      II1=1                                                             GEOM144   011960
      II2=NXX                                                           GEOM145   011970
      IF (IFSTA.NE.1) II2=IFSTA-1                                       GEOM146   011980
      IF (ILSTB.NE.NXX) II1=ILSTB+1                                     GEOM147   011990
      DO 130 II=II1,II2                                                 GEOM148   012000
      DO 130 JJ=1,NYY                                                   GEOM149   012010
  130 PERF(II,JJ)=0.                                                    GEOM150   012020
  140 IF (H.GT.EPSLN.OR.ABS(A)-90..GT..0001) GO TO 200                  GEOM151   012030
C                                                                       GEOM152   012040
C     PROJECTILE IMPACT ON PLATE                                        GEOM153   012050
      FNZ=NZ                                                            GEOM154   012060
      F=EPSLN/2.                                                        GEOM155   012070
      IF (H.GT.0.) F=SQRT(EPSLN**2-H**2)/2.                             GEOM156   012080
      IF (AINT(F).NE.F.OR.F.EQ.0.) F=F+1.                               GEOM157   012090
      K=F                                                               GEOM158   012100
      LIMYB=IROW-K                                                      GEOM159   012110
      LIMYT=IROW+K                                                      GEOM160   012120
      LIMXL=ICOL-K                                                      GEOM161   012130
      LIMXR=ICOL+K                                                      GEOM162   012140
      IF (LIMYB.LT.1) LIMYB=1                                           GEOM163   012150
      IF (LIMYT.GT.NYY) LIMYT=NYY                                       GEOM164   012160
      IF (LIMYB.GT.LIMYT) GO TO 200                                     GEOM165   012170
      IF (90.-ABS(A).LE..0001) GO TO 180                                GEOM166   012180
      TEMP1=EPSLN*COSA                                                  GEOM167   012190
      TEMP2=2.*FNZ*SINAA                                                GEOM168   012200
      IF (H.LT.TEMP1-TEMP2) GO TO 150                                   GEOM169   012210
      F=(TEMP1-H)/ABS(TANA)+EPSLN*SINAA                                 GEOM170   012220
      GO TO 170                                                         GEOM171   012230
  150 TEMP1=2.*FNZ*COSA                                                 GEOM172   012240
      IF (H.LT.-TEMP2) GO TO 160                                        GEOM173   012250
      F=TEMP1+SQRT(EPSLN**2-(TEMP2+H)**2)                               GEOM174   012260
      GO TO 170                                                         GEOM175   012270
  160 F=TEMP1+EPSLN                                                     GEOM176   012280
  170 F=F/2.                                                            GEOM177   012290
      IF (AINT(F).NE.F) F=F+1.                                          GEOM178   012300
      K=F                                                               GEOM179   012310
      IF (A.GT.0.) LIMXL=ICOL-K                                         GEOM180   012320
```

```
      IF (A.LT.O.) LIMXR=ICOL+K                                          GEOM181   012330
  180 IF (LIMXL.LT.1) LIMXL=1                                            GEOM182   012340
      IF (LIMXR.GT.NXX) LIMXR=NXX                                        GEOM183   012350
      IF (LIMXL.GT.LIMXR) GO TO 200                                      GEOM184   012360
C     ASSUME 10000. IS LARGE ENOUGH TO YIELD PK=1. FOR ALL KILL CRITERIAGEOM185   012370
      DO 190 J=LIMYB,LIMYT                                               GEOM186   012360
      DO 190 I=LIMXL,LIMXR                                               GEOM187   012390
  190 PERF(I,J)=10000.                                                   GEOM188   012400
C                                                                        GEOM189   012410
      IF (VOID2.EQ.0) GO TO 200                                          GEOM190   012420
      VOID2=0                                                            GEOM191   012430
      GO TO 210                                                          GEOM192   012440
  200 IF (VOID2.EQ.0) CALL BLANK                                         GEOM193   012450
  210 ICOL=ICOL-1                                                        GEOM194   012460
      GO TO 380                                                          GEOM195   012470
C                                                                        GEOM196   012480
C                                                                        GEOM197   012490
      ENTRY B1SID                                                        GEOM198E  012500
C     THIS ENTRY HANDLES A SINGLE SIDE WHERE THE JUNCTION OF THAT SIDE   GEOM199   012510
C     WITH THE TOP IS PARALLEL TO THE X AXIS. IN THIS ENTRY, BETA        GEOM200   012520
C     IS POSITIVE (NEGATIVE) WHEN THE OUTWARD DRAWN NORMAL TO THE SIDE   GEOM201   012530
C     HAS A COMPONENT IN THE POSITIVE (NEGATIVE) Y DIRECTION.            GEOM202   012540
C                                                                        GEOM203   012550
      IROW=IROW+1                                                        GEOM204   012560
      JFSTA=1                                                            GEOM205   012570
      JLSTB=NYY                                                          GEOM206   012580
      COSBB=COSB                                                         GEOM207   012590
      SINBB=SINB                                                         GEOM208   012600
      IF (B.GT.C.) GO TO 220                                             GEOM209   012610
      COSBB=-COSBB                                                       GEOM210   012620
      SINBB=-SINBB                                                       GEOM211   012630
  220 IF (NNSPRA.GT.C) GO TO 230                                         GEOM212   012640
      VOID2=1                                                            GEOM213   012650
      GO TO 300                                                          GEOM214   012660
  230 VAL=0.                                                             GEOM215   012670
      IF (ABS(ABS(B)-90.).GT..0001) VAL=H/(2.*TANB)                      GEOM216   012680
      IVAL=VAL                                                           GEOM217   012690
      F=IVAL                                                             GEOM218   012700
      IF (F.NE.VAL.AND.(90.-ABS(B).GT..0001.AND.H.GE.C..OR.ABS(B)-90..GTGEOM219   012710
     1..0001.AND.H.LT.C.)) GO TO 240                                     GEOM220   012720
      IVAL=IVAL-1                                                        GEOM221   012730
      IF (B.LT.O.) IVAL=IVAL+2                                           GEOM222   012740
  240 IF (B.GT.C..AND.IVAL+IROW.LT.NYY) JLSTB=IVAL+IROW                  GEOM223   012750
      IF (B.LT.O..AND.IVAL+IROW.GT.1) JFSTA=IVAL+IROW                    GEOM224   012760
      IF (JLSTB.GE.1.AND.JFSTA.LE.NYY) GO TO 250                         GEOM225   012770
      VOID2=2                                                            GEOM226   012780
      GO TO 300                                                          GEOM227   012790
C                                                                        GEOM228   012800
C     JJ SPECIFIES THE ROW BOUNDARY AT WHICH THE TOP EDGE OF THE SIDE ISGEOM229   012810
C     LOCATED. THE ROW BOUNDARY NUMBERED 'K' IS ON THE LOW ORDER SIDE    GEOM230   012820
C     OF ROW K.                                                          GEOM231   012830
  250 IF (H.GE.C..OR.ABS(B)-90..LT..0001) GO TO 260                      GEOM232   012840
      IF (JLSTB.GT.IROW.AND.B.GT.C.) JLSTB=IROW                          GEOM233   012850
      IF (JFSTA.LT.IROW.AND.B.LT.C.) JFSTA=IROW                          GEOM234   012860
      IF (JFSTA.LE.JLSTB) GO TO 260                                      GEOM235   012870
      VOID2=2                                                            GEOM236   012880
      GO TO 370                                                          GEOM237   012890
  260 DO 280 JJ=JFSTA,JLSTB                                              GEOM238   012900
      YY=2*(JJ-IROW)                                                     GEOM239   012910
      DO 280 II=1,NXX                                                    GEOM240   012920
```

AD-A068 976 ARMY MATERIEL SYSTEMS ANALYSIS ACTIVITY ABERDEEN PROV--ETC F/G 19/3
LIGHTLY ARMORED STRUCTURE VULNERABILITY ESTIMATION METHODOLOGY --ETC(U)
JAN 79 C J LAPOINTE
UNCLASSIFIED AMSAA-TR-254 NL

2 OF 3
AD
A068976




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
MICROCOPY RESOLUTION TEST CHART

```
      X=2*(II-ICOL)-1                                                   GEOM241  012930
      PRFK=0.                                                           GEOM242  012940
      DO 270 K=1,NZ                                                     GEOM243  012950
      FK=2*K-1                                                          GEOM244  012960
      Y=YY+FK*COSBB                                                     GEOM245  012970
      Z=-H-FK*SINBB                                                     GEOM246  012980
      R2=X**2+Y**2+Z**2                                                 GEOM247  012990
      R=SQRT(R2)                                                        GEOM248  013000
      ETA=-Z*COSB-Y*SINB                                                GEOM249  013010
      THETA=(X*COSW-Z*SINW)/R                                           GEOM250  013020
      STER=4.*ETA/(R2*R)                                                GEOM251  013030
C                                                                       GEOM252  013040
      CALL DAMGE                                                        GEOM253  013050
C                                                                       GEOM254  013060
  270 PRFK=PRFK+PRFI                                                    GEOM255  013070
  280 PERF(II,JJ)=PRFK/2.                                               GEOM256  013080
      IF (JFSTA.EQ.1.AND.JLSTB.EQ.NYY) GO TO 300                        GEOM257  013090
      JJ1=1                                                             GEOM258  013100
      JJ2=NYY                                                           GEOM259  013110
      IF (JFSTA.NE.1) JJ2=JFSTA-1                                       GEOM260  013120
      IF (JLSTB.NE.NYY) JJ1=JLSTB+1                                     GEOM261  013130
      DO 290 JJ=JJ1,JJ2                                                 GEOM262  013140
      DO 290 II=1,NXX                                                   GEOM263  013150
  290 PERF(II,JJ)=0.                                                    GEOM264  013160
  300 IF (H.GT.EPSLN.OR.ABS(B)-90..GT..0001) GO TO 360                  GEOM265  013170
C                                                                       GEOM266  013180
C     PROJECTILE IMPACT ON PLATE                                        GEOM267  013190
      FNZ=NZ                                                            GEOM268  013200
      F=EPSLN/2.                                                        GEOM269  013210
      IF (H.GT.0.) F=SQRT(EPSLN**2-H**2)/2.                             GEOM270  013220
      IF (AINT(F).NE.F.OR.F.EQ.0.) F=F+1.                               GEOM271  013230
      K=F                                                               GEOM272  013240
      LIMXL=ICOL-K                                                      GEOM273  013250
      LIMXR=ICOL+K                                                      GEOM274  013260
      LIMYB=IROW-K                                                      GEOM275  013270
      LIMYT=IROW+K                                                      GEOM276  013280
      IF (LIMXL.LT.1) LIMXL=1                                           GEOM277  013290
      IF (LIMXR.GT.NXX) LIMXR=NXX                                       GEOM278  013300
      IF (LIMXL.GT.LIMXR) GO TO 360                                     GEOM279  013310
      IF (90.-ABS(B).LE..0001) GO TO 340                                GEOM280  013320
      TEMP1=EPSLN*COSB                                                  GEOM281  013330
      TEMP2=2.*FNZ*SINBB                                                GEOM282  013340
      IF (H.LT.TEMP1-TEMP2) GO TO 310                                   GEOM283  013350
      F=(TEMP1-H)/ABS(TANB)+EPSLN*SINBB                                 GEOM284  013360
      GO TO 330                                                         GEOM285  013370
  310 TEMP1=2.*FNZ*COSB                                                 GEOM286  013380
      IF (H.LT.-TEMP2) GO TO 320                                        GEOM287  013390
      F=TEMP1+SQRT(EPSLN**2-(TEMP2+H)**2)                               GEOM288  013400
      GO TO 330                                                         GEOM289  013410
  320 F=TEMP1+EPSLN                                                     GEOM290  013420
  330 F=F/2.                                                            GEOM291  013430
      IF (AINT(F).NE.F) F=F+1.                                          GEOM292  013440
      K=F                                                               GEOM293  013450
      IF (B.GT.0.) LIMYB=IROW-K                                         GEOM294  013460
      IF (B.LT.0.) LIMYT=IROW+K                                         GEOM295  013470
  340 IF (LIMYB.LT.1) LIMYB=1                                           GEOM296  013480
      IF (LIMYT.GT.NYY) LIMYT=NYY                                       GEOM297  013490
      IF (LIMYB.GT.LIMYT) GO TO 360                                     GEOM298  013500
C     ASSUME 10000. IS LARGE ENOUGH TO YIELD PK=1. FOR ALL KILL CRITERIAGEOM299  013510
      DO 350 I=LIMXL,LIMXR                                              GEOM300  013520
```

```
      DO 350 J=LIMYB,LIMYT
  350 PERF(I,J)=10000.
C
      IF (VOID2.EQ.0) GO TO 360
      VOID2=0
      GO TO 370
  360 IF (VOID2.EQ.0) CALL BLANK
  370 IROW=IROW-1
  380 IF (VOID2.EQ.0) CALL SUMPF
      IF (VOID2.NE.0) CALL VOID
      RETURN
C
C
      ENTRY A2SID
C     THIS ENTRY HANDLES TWO SIDES WHERE THEIR JUNCTION WITH THE TOP IS
C     PARALLEL TO THE Y AXIS. IN THIS ENTRY, ALPHA AND FNX ARE ALWAYS
C     GREATER THAN ZERO.
C
C     HLFXA AND HLFXB DEFINE THE CENTER OF THE ASSEMBLY IN THE MANNER
C     USED BY SUBROUTINE SUMPF.
C     IFSTA IS THE FIRST II FOR WHICH SIDE A CONTRIBUTES. ILSTB IS THE
C     LAST II FOR WHICH SIDE B CONTRIBUTES.
C     SIDE A HAS A NORMAL WITH A COMPONENT IN THE -X DIRECTION AND
C     SIDE B, IN THE +X DIRECTION
      IF (NNSPRA.GT.0) GO TO 390
      VOID2=1
      GO TO 530
  390 ILFXA=2*(NX/2)+1
      HLFXA=ILFXA
      ILFXB=2*(NX-NX/2)-1
      HLFXB=ILFXB
      VAL=0.
      IF (ABS(A-90.).GT..0001) VAL=H/(2.*TANA)
      IVAL=VAL
      F=IVAL
      IF (F.NE.VAL.AND.(90.-A.GT..0001.AND.H.GE.0..OR.A-90..GT..0001.AND
     1.H.LT.0.)) GO TO 400
      IVAL=IVAL-1
  400 IFSTA=ICOL-IVAL+NX/2+1
      IF (IFSTA.LT.1) IFSTA=1
      ILSTB=ICOL+IVAL-NX+NX/2+1
      IF (ILSTB.GT.NXX) ILSTB=NXX
      IF (IFSTA.LE.NXX.OR.ILSTB.GE.1) GO TO 410
      VOID2=2
      GO TO 530
C
C     ONLY B CONTRIBUTES
  410 ISTOP=IFSTA-1
      IF (IFSTA.GT.ILSTB) ISTOP=ILSTB
      IF (H.LT.0..AND.ISTOP.GT.ICOL-NX+NX/2+1) ISTOP=ICOL-NX+NX/2+1
      IF (ISTOP.LT.1) GO TO 440
C     LOOP ON X COORDINATE OF TARGET CENTER
      DO 430 II=1,ISTOP
      XXB=2*(II-ICOL)-1+ILFXB
C     LOOP ON Y COORDINATE OF STRIP OF CELLS IN SLAB
      DO 430 JJ=1,NYY
      Y=2*(JJ-IROW)-1
C     LOOP ON ALL CELLS IN STRIP
      PRFK=0.
      DO 420 K=1,NZ
```

```
      FK=2*K-1                                                         GEOM361  014130
      XB=XXB+FK*COSA                                                   GEOM362  014140
      Z=-H-FK*SINA                                                     GEOM363  014150
      RB2=XB**2+Y**2+Z**2                                              GEOM364  014160
      RB=SQRT(RB2)                                                     GEOM365  014170
C     RB*COS(ETA), ETA=ANGLE BETWEEN RB AND CELL NORMAL                GEOM366  014180
      ETA=-Z*COSA-XB*SINA                                              GEOM367  014190
      THETA=(XB*COSW-Z*SINW)/RB                                        GEOM368  014200
      STER=4.*ETA/(RB2*RB)                                             GEOM369  014210
C     GEOMETRY FOR THIS CELL IS COMPLETE                               GEOM370  014220
C                                                                      GEOM371  014230
      CALL DAMGE                                                       GEOM372  014240
C                                                                      GEOM373  014250
  420 PRFK=PRFK+PRF1                                                   GEOM374  014260
  430 PERF(II,JJ)=PRFK/2.                                              GEOM375  014270
C     REGION IN WHICH ONLY B CONTRIBUTES IS DONE                       GEOM376  014280
C                                                                      GEOM377  014290
C     BOTH A AND B CONTRIBUTE                                          GEOM378  014300
  440 IF (H.LT.0.) GO TO 450                                           GEOM379  014310
      IF (IFSTA.LE.ILSTB) GO TO 470                                    GEOM380  014320
  450 I1=ILSTB+1                                                       GEOM381  014330
      IF (I1.LT.1) I1=1                                                GEOM382  014340
      IF (IFSTA.GT.NXX+1) IFSTA=NXX+1                                  GEOM383  014350
      IF (I1.EQ.IFSTA) GO TO 500                                       GEOM384  014360
      I2=IFSTA-1                                                       GEOM385  014370
      DO 460 II=I1,I2                                                  GEOM386  014380
      DO 460 JJ=1,NYY                                                  GEOM387  014390
  460 PERF(II,JJ)=0.                                                   GEOM388  014400
      GO TO 500                                                        GEOM389  014410
  470 DO 490 II=IFSTA,ILSTB                                            GEOM390  014420
      XX=2*(II-ICOL)-1                                                 GEOM391  014430
      XXA=XX-HLFXA                                                     GEOM392  014440
      XXB=XX+HLFXB                                                     GEOM393  014450
      DO 490 JJ=1,NYY                                                  GEOM394  014460
      Y=2*(JJ-IROW)-1                                                  GEOM395  014470
      PRFK=0.                                                          GEOM396  014480
      DO 480 K=1,NZ                                                    GEOM397  014490
      FK=2*K-1                                                         GEOM398  014500
      XB=XXB+FK*COSA                                                   GEOM399  014510
      XA=XXA-FK*COSA                                                   GEOM400  014520
      Z=-H-FK*SINA                                                     GEOM401  014530
      YZ2=Y**2+Z**2                                                    GEOM402  014540
      RB2=XB**2+YZ2                                                    GEOM403  014550
      RA2=XA**2+YZ2                                                    GEOM404  014560
C     SIDE B                                                           GEOM405  014570
      RB=SQRT(RB2)                                                     GEOM406  014580
      ETA=-Z*COSA-XB*SINA                                              GEOM407  014590
      THETA=(XB*COSW-Z*SINW)/RB                                        GEOM408  014600
      STER=4.*ETA/(RB2*RB)                                             GEOM409  014610
C                                                                      GEOM410  014620
      CALL DAMGE                                                       GEOM411  014630
C                                                                      GEOM412  014640
      PRFK=PRFK+PRF1                                                   GEOM413  014650
C     SIDE A                                                           GEOM414  014660
      RA=SQRT(RA2)                                                     GEOM415  014670
      ETA=XA*SINA-Z*COSA                                               GEOM416  014680
      THETA=(XA*COSW-Z*SINW)/RA                                        GEOM417  014690
      STER=4.*ETA/(RA2*RA)                                             GEOM418  014700
C                                                                      GEOM419  014710
      CALL DAMGE                                                       GEOM420  014720
```

```
C                                                                       GEOM421  C14730
  480 PRFK=PRFK+PRF1                                                    GEOM422  C14740
  490 PERF(II,JJ)=PRFK/2.                                               GEOM423  014750
C     THIS FINISHES THE REGION IN WHICH BOTH A AND B CONTRIBUTE.        GEOM424  014760
C                                                                       GEOM425  014770
C     ONLY A CONTRIBUTES                                                GEOM426  014780
  500 ISTRT=ILSTB+1                                                     GEOM427  014790
      IF (IFSTA.GT.ILSTB) ISTRT=IFSTA                                   GEOM428  014800
      IF (H.LT.0..AND.ISTRT.LT.ICOL+NX/2+1) ISTRT=ICOL+NX/2+1           GEOM429  014810
      IF (ISTRT.GT.NXX) GO TO 530                                       GEOM430  014820
      DO 520 II=ISTRT,NXX                                               GEOM431  014830
      XXA=2*(II-ICOL)-1-ILFXA                                           GEOM432  014840
      DO 520 JJ=1,NYY                                                   GEOM433  014850
      Y=2*(JJ-IROW)-1                                                   GEOM434  014860
      PRFK=0.                                                           GEOM435  014870
      DO 510 K=1,NZ                                                     GEOM436  014880
      FK=2*K-1                                                          GEOM437  014890
      XA=XXA-FK*COSA                                                    GEOM438  014900
      Z=-H-FK*SINA                                                      GEOM439  014910
      RA2=XA**2+Y**2+Z**2                                               GEOM440  014920
      RA=SQRT(RA2)                                                      GEOM441  014930
      ETA=XA*SINA-Z*COSA                                                GEOM442  014940
      THETA=(XA*COSW-Z*SINW)/RA                                         GEOM443  014950
      STER=4.*ETA/(RA2*RA)                                              GEOM444  014960
C                                                                       GEOM445  014970
      CALL DAMGE                                                        GEOM446  014980
C                                                                       GEOM447  C14990
  510 PRFK=PRFK+PRF1                                                    GEOM448  015000
  520 PERF(II,JJ)=PRFK/2.                                               GEOM449  015010
C                                                                       GEOM450  015020
  530 IF (H.GT.EPSLN.OR.A-90..GT..0001) GO TO 880                       GEOM451  015030
      IF (VOID2.EQ.0) GO TO 550                                         GEOM452  015040
      DO 540 I=1,NXX                                                    GEOM453  015050
      DO 540 J=1,NYY                                                    GEOM454  015060
  540 PERF(I,J)=0.                                                      GEOM455  015070
C                                                                       GEOM456  015080
C     PROJECTILE IMPACT ON SIDES                                        GEOM457  015090
  550 FNZ=NZ                                                            GEOM458  015100
      F=EPSLN/2.                                                        GEOM459  015110
      IF (H.GT.0.) F=SQRT(EPSLN**2-H**2)/2.                             GEOM460  015120
      IF (AINT(F).NE.F.OR.F.EQ.0.) F=F+1.                               GEOM461  015130
      K=F                                                               GEOM462  015140
      JSTRT=IROW-K                                                      GEOM463  015150
      JSTOP=IROW+K                                                      GEOM464  015160
      ISTOP=ICOL+K-NX+NX/2+1                                            GEOM465  015170
      KSTRT=ICOL-K+NX/2+1                                               GEOM466  015180
      IF (JSTRT.LT.1) JSTRT=1                                           GEOM467  015190
      IF (JSTOP.GT.NYY) JSTOP=NYY                                       GEOM468  015200
      IF (JSTRT.GT.JSTOP) GO TO 880                                     GEOM469  015210
      IF (90.-A.LE..0001) GO TO 590                                     GEOM470  015220
      TEMP1=EPSLN*COSA                                                  GEOM471  015230
      TEMP2=2.*FNZ*SINA                                                 GEOM472  015240
      IF (H.LT.TEMP1-TEMP2) GO TO 560                                   GEOM473  015250
      F=(TEMP1-H)/TANA+EPSLN*SINA                                       GEOM474  015260
      GO TO 580                                                         GEOM475  015270
  560 TEMP1=2.*FNZ*COSA                                                 GEOM476  015280
      IF (H.LT.-TEMP2) GO TO 570                                        GEOM477  015290
      F=TEMP1+SQRT(EPSLN**2-(TEMP2+H)**2)                               GEOM478  015300
      GO TO 580                                                         GEOM479  015310
  570 F=TEMP1+EPSLN                                                     GEOM480  015320
```

```
580 F=F/2.                                                          GEOM481   015330
    IF (AINT(F).NE.F) F=F+1.                                        GEOM482   015340
    K=F                                                             GEOM483   015350
590 ISTRT=ICOL-K-NX+NX/2+1                                          GEOM484   015360
    IF (ISTRT.LT.1) ISTRT=1                                         GEOM485   015370
    IF (ISTOP.GT.NXX) ISTOP=NXX                                     GEOM486   015380
    KSTOP=ICOL+K+NX/2+1                                             GEOM487   015390
    IF (KSTRT.LT.1) KSTRT=1                                         GEOM488   015400
    IF (KSTOP.GT.NXX) KSTOP=NXX                                     GEOM489   015410
C   ASSUME 10000. IS LARGE ENOUGH TO YIELD PK=1. FOR ALL KILL CRITERIAGEOM490   015420
    IF (ISTRT.GT.ISTOP) GO TO 610                                   GEOM491   015430
C   SIDE B                                                          GEOM492   015440
    DO 600 J=JSTRT,JSTOP                                            GEOM493   015450
    DO 600 I=ISTRT,ISTOP                                            GEOM494   015460
600 PERF(I,J)=10000.                                                GEOM495   015470
610 IF (KSTRT.GT.KSTOP) GO TO 860                                   GEOM496   015480
C   SIDE A                                                          GEOM497   015490
    DO 620 J=JSTRT,JSTOP                                            GEOM498   015500
    DO 620 K=KSTRT,KSTOP                                            GEOM499   015510
620 PERF(K,J)=10000.                                                GEOM500   015520
    GO TO 870                                                       GEOM501   015530
C                                                                   GEOM502   015540
C                                                                   GEOM503   015550
    ENTRY B2SID                                                     GEOM504E  015560
C   THIS ENTRY HANDLES TWO SIDES WHERE THEIR JUNCTION WITH THE TOP IS GEOM505   015570
C   PARALLEL TO THE X AXIS. IN THIS ENTRY, BETA AND FNY ARE ALWAYS  GEOM506   015580
C   GREATER THAN ZERO.                                              GEOM507   015590
C                                                                   GEOM508   015600
C   HLFYA AND HLFYB DEFINE THE CENTER OF THE ASSEMBLY IN THE MANNER GEOM509   015610
C   USED BY SUBROUTINE SUMPF.                                       GEOM510   015620
C   JFSTA IS THE FIRST JJ FOR WHICH SIDE A CONTRIBUTES. JLSTB IS THE GEOM511   015630
C   LAST JJ FOR WHICH SIDE B CONTRIBUTES.                           GEOM512   015640
C   SIDE A HAS A NORMAL WITH A COMPONENT IN THE -X DIRECTION AND    GEOM513   015650
C   SIDE B, IN THE +X DIRECTION                                     GEOM514   015660
    IF (NNSPRA.GT.0) GO TO 630                                      GEOM515   015670
    VOID2=1                                                         GEOM516   015680
    GO TO 770                                                       GEOM517   015690
630 ILFYA=2*(NY/2)+1                                                GEOM518   015700
    HLFYA=ILFYA                                                     GEOM519   015710
    ILFYB=2*(NY-NY/2)-1                                             GEOM520   015720
    HLFYB=ILFYB                                                     GEOM521   015730
    VAL=0.                                                          GEOM522   015740
    IF (ABS(B-90.).GT..0001) VAL=H/(2.*TANB)                        GEOM523   015750
    IVAL=VAL                                                        GEOM524   015760
    F=IVAL                                                          GEOM525   015770
    IF (F.NE.VAL.AND.(90.-B.GT..0001.AND.H.GE.0..OR.B-90..GT..0001.ANDGEOM526   015780
   1.H.LT.0.)) GO TO 640                                            GEOM527   015790
    IVAL=IVAL-1                                                     GEOM528   015800
640 JFSTA=IROW-IVAL+NY/2+1                                          GEOM529   015810
    IF (JFSTA.LT.1) JFSTA=1                                         GEOM530   015820
    JLSTB=IROW+IVAL-NY+NY/2+1                                       GEOM531   015830
    IF (JLSTB.GT.NYY) JLSTB=NYY                                     GEOM532   015840
C                                                                   GEOM533   015850
    IF (JFSTA.LE.NYY.OR.JLSTB.GE.1) GO TO 650                       GEOM534   015860
    VOID2=2                                                         GEOM535   015870
    GO TO 770                                                       GEOM536   015880
C   ONLY B CONTRIBUTES                                              GEOM537   015890
650 JSTOP=JFSTA-1                                                   GEOM538   015900
    IF (JFSTA.GT.JLSTB) JSTOP=JLSTB                                 GEOM539   015910
    IF (H.LT.0..AND.JSTOP.GT.IROW-NY+NY/2+1) JSTOP=IROW-NY+NY/2+1   GEOM540   015920
```

```
      IF (JSTOP.LT.1) GO TO 660                                          GEOM541  015930
C     LOOP ON Y COORDINATE OF TARGET CENTER                              GEOM542  015940
      DO 670 JJ=1,JSTOP                                                  GEOM543  015950
      YYB=2*(JJ-IROW)-1+ILFYB                                            GEOM544  015960
C     LOOP ON X COORDINATE OF STRIP OF CELLS IN SLAB                     GEOM545  015970
      DO 670 II=1,NXX                                                    GEOM546  015980
      X=2*(II-ICOL)-1                                                    GEOM547  015990
C     LOOP ON ALL CELLS IN STRIP                                         GEOM548  016000
      PRFK=0.                                                            GEOM549  016010
      DO 660 K=1,NZ                                                      GEOM550  016020
      FK=2*K-1                                                           GEOM551  016030
      YB=YYB+FK*COSB                                                     GEOM552  016040
      Z=-H-FK*SINB                                                       GEOM553  016050
      RB2=YB**2+X**2+Z**2                                                GEOM554  016060
      RB=SQRT(RB2)                                                       GEOM555  016070
C     RB*COS(ETA), ETA=ANGLE BETWEEN RB AND CELL NORMAL                  GEOM556  016080
      ETA=-Z*COSB-YB*SINB                                                GEOM557  016090
      THETA=(X*COSW-Z*SINW)/RB                                           GEOM558  016100
      STER=4.*ETA/(RB2*RB)                                               GEOM559  016110
C     GEOMETRY FOR THIS CELL IS COMPLETE                                 GEOM560  016120
C                                                                        GEOM561  016130
      CALL DAMGE                                                         GEOM562  016140
C                                                                        GEOM563  016150
  660 PRFK=PRFK+PRF1                                                     GEOM564  016160
  670 PERF(II,JJ)=PRFK/2.                                                GEOM565  016170
C     REGION IN WHICH ONLY B CONTRIBUTES IS DONE                         GEOM566  016180
C                                                                        GEOM567  016190
C     BOTH A AND B CONTRIBUTE                                            GEOM568  016200
  680 IF (H.LT.C.) GO TO 690                                             GEOM569  016210
      IF (JFSTA.LE.JLSTB) GO TO 710                                      GEOM570  016220
  690 J1=JLSTB+1                                                         GEOM571  016230
      IF (J1.LT.1) J1=1                                                  GEOM572  016240
      IF (JFSTA.GT.NYY+1) JFSTA=NYY+1                                    GEOM573  016250
      IF (J1.EQ.JFSTA) GO TO 740                                         GEOM574  016260
      J2=JFSTA-1                                                         GEOM575  016270
      DO 700 JJ=J1,J2                                                    GEOM576  016280
      DO 700 II=1,NXX                                                    GEOM577  016290
  700 PERF(II,JJ)=C.                                                     GEOM578  016300
      GO TO 740                                                          GEOM579  016310
  710 DO 730 JJ=JFSTA,JLSTB                                              GEOM580  016320
      YY=2*(JJ-IROW)-1                                                   GEOM581  016330
      YYA=YY-HLFYA                                                       GEOM582  016340
      YYB=YY+HLFYB                                                       GEOM583  016350
      DO 730 II=1,NXX                                                    GEOM584  016360
      X=2*(II-ICOL)-1                                                    GEOM585  016370
      PRFK=C.                                                            GEOM586  016380
      DO 720 K=1,NZ                                                      GEOM587  016390
      FK=2*K-1                                                           GEOM588  016400
      YB=YYB+FK*COSB                                                     GEOM589  016410
      YA=YYA-FK*COSB                                                     GEOM590  016420
      Z=-H-FK*SINB                                                       GEOM591  016430
      XZ2=X**2+Z**2                                                      GEOM592  016440
      RB2=YB**2+XZ2                                                      GEOM593  016450
      RA2=YA**2+XZ2                                                      GEOM594  016460
      THET=X*COSW-Z*SINW                                                 GEOM595  016470
C     SIDE B                                                             GEOM596  016480
      RB=SQRT(RB2)                                                       GEOM597  016490
      ETA=-Z*COSB-YB*SINB                                                GEOM598  016500
      THETA=THET/RB                                                      GEOM599  016510
      STER=4.*ETA/(RB2*RB)                                               GEOM600  016520
```

```
C
      CALL DAMGE
C
      PRFK=PRFK+PRF1
C     SIDE A
      RA=SQRT(RA2)
      ETA=YA*SINB-Z*COSB
      THETA=THET/RA
      STER=4.*ETA/(RA2*RA)
C
      CALL DAMGE
C
  720 PRFK=PRFK+PRF1
  730 PERF(II,JJ)=PRFK/2.
C     THIS FINISHES THE REGION IN WHICH BOTH A AND B CONTRIBUTE.
C
C     ONLY A CONTRIBUTES
  740 JSTRT=JLSTB+1
      IF (JFSTA.GT.JLSTB) JSTRT=JFSTA
      IF (H.LT.0..AND.JSTRT.LT.IROW+NY/2+1) JSTRT=IROW+NY/2+1
      IF (JSTRT.GT.NYY) GO TO 770
      DO 760 JJ=JSTRT,NYY
      YYA=2*(JJ-IROW)-1-ILFYA
      DO 760 II=1,NXX
      X=2*(II-ICOL)-1
      PRFK=C.
      DO 750 K=1,NZ
      FK=2*K-1
      YA=YYA-FK*COSB
      Z=-H-FK*SINB
      RA2=YA**2+X**2+Z**2
      RA=SQRT(RA2)
      ETA=YA*SINB-Z*COSB
      THETA=(X*COSW-Z*SINW)/RA
      STER=4.*ETA/(RA2*RA)
C
      CALL DAMGE
C
  750 PRFK=PRFK+PRF1
  760 PERF(II,JJ)=PRFK/2.
C
  770 IF (H.GT.EPSLN.OR.B-90..GT..C001) GO TO 880
      IF (VOID2.EQ.0) GO TO 790
      DO 780 I=1,NXX
      DO 780 J=1,NYY
  780 PERF(I,J)=C.
C
C     PROJECTILE IMPACT ON SIDES
  790 FNZ=NZ
      F=EPSLN/2.
      IF (H.GT.C.) F=SQRT(EPSLN**2-H**2)/2.
      IF (AINT(F).NE.F.OR.F.EQ.C.) F=F+1.
      K=F
      ISTRT=ICOL-K
      ISTOP=ICCL+K
      JSTCP=IROW+K-NY+NY/2+1
      KSTRT=IROW-K+NY/2+1
      IF (ISTRT.LT.1) ISTRT=1
      IF (ISTCP.GT.NXX) ISTOP=NXX
      IF (ISTRT.GT.ISTOP) GO TO 88C
```

```
      IF (90.-B.LE..COC1) GO TO 830                                       GEOM661  017130
      TEMP1=EPSLN*COSB                                                    GEOM662  017140
      TEMP2=2.*FNZ*SINB                                                   GEOM663  017150
      IF (H.LT.TEMP1-TEMP2) GO TO 800                                     GEOM664  017160
      F=(TEMP1-H)/TANB+EPSLN*SINB                                         GEOM665  017170
      GO TO 820                                                           GEOM666  017180
  800 TEMP1=2.*FNZ*COSB                                                   GEOM667  017190
      IF (H.LT.-TEMP2) GO TO 810                                          GEOM668  017200
      F=TEMP1+SQRT(EPSLN**2-(TEMP2+H)**2)                                 GEOM669  017210
      GO TO 820                                                           GEOM670  017220
  810 F=TEMP1+EPSLN                                                       GEOM671  017230
  820 F=F/2.                                                              GEOM672  017240
      IF (AINT(F).NE.F) F=F+1.                                            GEOM673  017250
      K=F                                                                 GEOM674  017260
  830 JSTRT=IROW-K-NY+NY/2+1                                              GEOM675  017270
      KSTOP=IROW+K+NY/2+1                                                 GEOM676  017280
      IF (JSTRT.LT.1) JSTRT=1                                             GEOM677  017290
      IF (JSTOP.GT.NYY) JSTOP=NYY                                         GEOM678  017300
      IF (KSTRT.LT.1) KSTRT=1                                             GEOM679  017310
      IF (KSTOP.GT.NYY) KSTOP=NYY                                         GEOM680  017320
C     ASSUME 10000. IS LARGE ENOUGH TO YIELD PK=1. FOR ALL KILL CRITERIAGEOM681  017330
      IF (JSTRT.GT.JSTOP) GO TO 850                                       GEOM682  017340
C     SIDE B                                                              GEOM683  017350
      DO 840 I=ISTRT,ISTOP                                                GEOM684  017360
      DO 840 J=JSTRT,JSTOP                                                GEOM685  017370
  840 PERF(I,J)=10000.                                                    GEOM686  017380
  850 IF (KSTRT.GT.KSTOP) GO TO 860                                       GEOM687  017390
C     SIDE A                                                              GEOM688  017400
      DO 860 I=ISTRT,ISTOP                                                GEOM689  017410
      DO 860 K=KSTRT,KSTOP                                                GEOM690  017420
  860 PERF(I,K)=10000.                                                    GEOM691  017430
C                                                                         GEOM692  017440
C     ALL LOCATIONS OF THE SLAB (PAIR) ARE DONE                           GEOM693  017450
  870 VOID2=0                                                             GEOM694  017460
  880 IF (VOID2.NE.0) GO TO 910                                           GEOM695  017470
  890 CALL BLANK                                                          GEOM696  017480
  900 IF (VOID2.NE.0) GO TO 910                                           GEOM697  017490
      CALL SUMPF                                                          GEOM698  017500
      RETURN                                                              GEOM699  017510
  910 CALL VOID                                                           GEOM700  017520
      RETURN                                                              GEOM701  017530
      END                                                                 GEOM702- 017540
```

```
      SUBROUTINE GOTHRU                                                 *1745* 7 017550
C     THIS SUBROUTINE IDENTIFIES THOSE FRAGMENTS WHICH HAVE THE          GOTHU 2  017560
C     CAPABILITY OF PERFORATING A PLATE OF THE CURRENT THICKNESS AT 0    GOTHU 3  017570
C     SEPARATION FROM THE FRAGMENT SOURCE AND WITH 0 OBLIQUITY. IT SETS  GOTHU 4  017580
C     UP THE TABLE USED BY DAMAGE TO PICK OUT THOSE FRAGMENTS. IT        GOTHU 5  017590
C     DEMANDS THAT THE Q AND M TABLES BE ORDERED WITH THE LEAST MASSIVE  GOTHU 6  017600
C     FRAGMENTS FIRST.                                                   GOTHU 7  017610
C                                                                        GOTHU 8  017620
      COMMON /COM3/COM3(10000)                                           GOTHU 9  017630
      DIMENSION VDYN(92),NSP(46),M(40,46),SGST(46),XREF(3680)            GOTHU10  017640
      EQUIVALENCE  (COM3(231),VDYN(1)),(COM3(323),NSP(1))                GOTHU11  017650
     1            ,(COM3(369),M(1,1)),(COM3(4049),SGST(1))               GOTHU12  017660
     2            ,(COM3(4713),XREF(1)),(COM3(9939),NNSPRA)              GOTHU13  017670
     3            ,(COM3(9952),LAMDAM),(COM3(9954),CNST3V)               GOTHU14  017680
     4            ,(COM3(9955),CNST4V),(COM3(9958),CNST4M)               GOTHU15  017690
     5            ,(COM3(9983),K22),(COM3(9984),K222)                    GOTHU16  017700
C                                                                        GOTHU17  017710
      COMMON /THOR70/RATIOM,RATIOV                                       GOTHU18  017720
C                                                                        GOTHU19  017730
      INTEGER XREF                                                       GOTHU20  017740
      REAL   LAMDAM,M,MLOSS                                              GOTHU21  017750
C                                                                        GOTHU22  017760
      NNSPRA=0                                                           GOTHU23  017770
C     LOOP ON THE SPRAYS                                                 GOTHU24  017780
      DO 50 K2=K22,K222                                                  GOTHU25  017790
      IF (SGST(K2).EQ.0.) GO TO 50                                       GOTHU26  017800
      ASSIGN 70 TO JUMP1                                                 GOTHU27  017810
      ASSIGN 10 TO JUMP2                                                 GOTHU28  017820
C     START WITH MOST MASSIVE FRAGMENTS                                  GOTHU29  017830
      K1=NSP(K2)                                                         GOTHU30  017840
C     DO VDYMN FIRST                                                     GOTHU31  017850
   10 VLOSS=CNST4V+CNST3V*M(K1,K2)+RATIOV                                GOTHU32  017860
      IF (VLOSS.GE.VDYN(K2)) GO TO 60                                    GOTHU33  017870
      MLOSS=CNST4M+LAMDAM*VDYN(K2)+RATIOM                                GOTHU34  017880
      IF (MLOSS.GE.M(K1,K2)) GO TO 50                                    GOTHU35  017890
C     PERFORATES                                                         GOTHU36  017900
      NNSPRA=NNSPRA+1                                                    GOTHU37  017910
      CALL PUT (XREF(NNSPRA),K1,K2,K2)                                   GOTHU38  017920
C     DO VDYMX                                                           GOTHU39  017930
   20 K3=K2+46                                                           GOTHU40  017940
   30 VLOSS=CNST4V+CNST3V*M(K1,K2)+RATIOV                                GOTHU41  017950
      IF (VLOSS.GE.VDYN(K3)) GO TO 50                                    GOTHU42  017960
      MLOSS=CNST4M+LAMDAM*VDYN(K3)+RATIOM                                GOTHU43  017970
      IF (MLOSS.GE.M(K1,K2)) GO TO JUMP1, (50,70)                        GOTHU44  017980
C     PERFORATES                                                         GOTHU45  017990
      NNSPRA=NNSPRA+1                                                    GOTHU46  018000
      CALL PUT (XREF(NNSPRA),K1,K2,K3)                                   GOTHU47  018010
C     NEXT SMALLER MASS, THIS SPRAY                                      GOTHU48  018020
   40 K1=K1-1                                                            GOTHU49  018030
      IF (K1.LT.1) GO TO 50                                              GOTHU50  018040
      GO TO JUMP2, (10,30)                                               GOTHU51  018050
   50 CONTINUE                                                           GOTHU52  018060
      RETURN                                                             GOTHU53  018070
C                                                                        GOTHU54  018080
   60 ASSIGN 50 TO JUMP1                                                 GOTHU55  018090
      ASSIGN 30 TO JUMP2                                                 GOTHU56  018100
      GO TO 20                                                           GOTHU57  018110
C                                                                        GOTHU58  018120
   70 ASSIGN 50 TO JUMP1                                                 GOTHU59  018130
      ASSIGN 30 TO JUMP2                                                 GOTHU60  018140
```

```
      K3=K2                                                             GOTHU61  018150
      GO TO 40                                                          GOTHU62  018160
      END                                                               GOTHU63- 018170

      SUBROUTINE PUT (K,K1,K2,K3)                                       *1808* 8 018180
C     THIS SUBROUTINE PACKS AND RETRIEVES THE 3 INTEGERS K1, K2, AND K3 PUT   2  018190
C     INTO AND FROM WORD K. IN PRACTICE, WORD K IS AN ELEMENT OF THE    PUT   3  018200
C     XREF ARRAY WHICH REFERENCES THOSE FRAGMENTS IDENTIFIED BY         PUT   4  018210
C     SUBROUTINE GOTHRU AS CAPABLE OF PERFORATING THE SUBJECT PLATE.    PUT   5  018220
C                                                                       PUT   6  018230
      K=SHIFT(SHIFT(K1.AND..NOT.MASK(40),20).OR.(K2.AND..NOT.MASK(40)   PUT   7  018240
     1  ),20).OR.(K3.AND..NOT.MASK(40))                                 PUT   8  018250
      GO TO 10                                                          PUT   9  018260
C                                                                       PUT  10  018270
      ENTRY GET                                                         PUT  11E 018280
      K1=SHIFT(K,-40).AND..NOT.MASK(40)                                 PUT  12  018290
      K2=SHIFT(K,-20).AND..NOT.MASK(40)                                 PUT  13  018300
      K3=K.AND..NOT.MASK(40)                                            PUT  14  018310
   10 RETURN                                                            PUT  15  018320
      END                                                               PUT  16- 018330
```

```
      SUBROUTINE DAMGE                                                  *1624* 9 018340
C     THIS SUBROUTINE GETS THE PERFORATIONS ON EACH CELL. M AND VDYN    DAMGE 2  018350
C     ARE THEIR OWN LOGARITHMS. VIMPCT IS THE (LOGARITHM OF THE)        DAMGE 3  018360
C     VELOCITY WITH WHICH THE FRAGMENT STRIKES THE PLATE.               DAMGE 4  018370
C                                                                       DAMGE 5  018380
      COMMON /COM3/COM3(10000)                                          DAMGE 6  018390
      DIMENSION SPRBD(92),VDYN(92),M(40,46),Q(40,46),SGDIN(92)          DAMGE 7  018400
     1          ,XREF(3680)                                             DAMGE 8  018410
      EQUIVALENCE  (COM3(47),SPRBD(1)),(COM3(231),VDYN(1))              DAMGE 9  018420
     1            ,(COM3(369),M(1,1)),(COM3(2209),Q(1,1))               DAMGE10  018430
     2            ,(COM3(4095),SGDIN(1)),(COM3(4713),XREF(1))           DAMGE11  018440
     3            ,(COM3(9939),NNSPRA),(COM3(9946),GAMAV)               DAMGE12  018450
     4            ,(COM3(9951),GAMAM),(COM3(9952),LAMDAM)               DAMGE13  018460
     5            ,(COM3(9953),CNST1),(COM3(9954),CNST3V)               DAMGE14  018470
     6            ,(COM3(9955),CNST4V),(COM3(9956),CNST5V)              DAMGE15  018480
     7            ,(COM3(9958),CNST4M),(COM3(9959),CNST5M)              DAMGE16  018490
     8            ,(COM3(9993),R),(COM3(9994),THETA),(COM3(9995),STER)  DAMGE17  018500
     9            ,(COM3(9996),PRF1),(COM3(9999),ETA)                   DAMGE18  018510
C                                                                       DAMGE19  018520
      COMMON /THOR70/RATIOM,RATIOV                                      DAMGE20  018530
C                                                                       DAMGE21  018540
      INTEGER XREF                                                      DAMGE22  018550
      REAL    LAMDAM,M,MLOSS                                            DAMGE23  018560
C                                                                       DAMGE24  018570
      DATA TEN /.46415688834/                                           DAMGE25D 018580
C     TEN=10.**(-1./3.)                                                 DAMGE26  018590
C                                                                       DAMGE27  018600
      PRF1=0.                                                           DAMGE28  018610
      TEMP3=ALOG10(R/ETA)                                               DAMGE29  018620
      CNST5V=CNST4V+GAMAV*TEMP3                                         DAMGE30  018630
      CNST5M=CNST4M+GAMAM*TEMP3                                         DAMGE31  018640
      CNST6=CNST1*R                                                     DAMGE32  018650
C                                                                       DAMGE33  018660
      KI=0                                                              DAMGE34  018670
   10 KI=KI+1                                                           DAMGE35  018680
      CALL GET (XREF(KI),K1,K2,K3)                                      DAMGE36  018690
   20 IF (THETA.GT.SPRBD(K3).OR.THETA.LE.SPRBD(K3+1)) GO TO 40          DAMGE37  018700
      VIMPCT=VDYN(K3)+CNST6*TEN**M(K1,K2)                               DAMGE38  018710
      VLOSS=CNST5V+CNST3V*M(K1,K2)+RATIOV                               DAMGE39  018720
      MLOSS=CNST5M+LAMDAM*VIMPCT+RATIOM                                 DAMGE40  018730
      IF (VLOSS.GE.VIMPCT.OR.MLOSS.GE.M(K1,K2)) GO TO 40                DAMGE41  018740
      PRF1=PRF1+SGDIN(K3)*STER*Q(K1,K2)                                 DAMGE42  018750
      IF (KI.LT.NNSPRA) GO TO 10                                        DAMGE43  018760
   30 RETURN                                                            DAMGE44  018770
C                                                                       DAMGE45  018780
   40 IF (KI.EQ.NNSPRA) GO TO 30                                        DAMGE46  018790
      KI1=KI                                                            DAMGE47  018800
      KI=KI+1                                                           DAMGE48  018810
   50 KI1=KI1+1                                                         DAMGE49  018820
      CALL GET (XREF(KI1),K1,K2,K31)                                    DAMGE50  018830
      IF (K31.EQ.K3) GO TO 60                                           DAMGE51  018840
      K3=K31                                                            DAMGE52  018850
      KI=KI1                                                            DAMGE53  018860
      GO TO 20                                                          DAMGE54  018870
   60 IF (KI1.LT.NNSPRA) GO TO 50                                       DAMGE55  018880
      GO TO 30                                                          DAMGE56  018890
      END                                                               DAMGE57- 018900
```

```
      SUBROUTINE BLANK                                                  *1681*10 018910
C     THIS SUBROUTINE ELIMINATES FROM CONSIDERATION THAT PORTION OF THE BLANK 2  018920
C     PERF ARRAY WHICH IS VOID                                          BLANK 3  018930
C                                                                       BLANK 4  018940
      COMMON /COM1/COM1(20100)                                          BLANK 5  018950
      DIMENSION PERF(100,100)                                           BLANK 6  018960
      EQUIVALENCE (COM1(1),PERF(1,1))                                   BLANK 7  018970
C                                                                       BLANK 8  018980
      COMMON /COM3/COM3(10000)                                          BLANK 9  018990
      EQUIVALENCE  (COM3(9979),LIMXL),(COM3(9980),LIMXR)                BLANK10  019000
     1             ,(COM3(9981),LIMYB),(COM3(9982),LIMYT)               BLANK11  019010
C                                                                       BLANK12  019020
      COMMON /NXXNYY/NXX,NYY                                            BLANK13  019030
      COMMON /VOID2/VOID2                                               BLANK14  019040
C                                                                       BLANK15  019050
      INTEGER VOID2                                                     BLANK16  019060
C                                                                       BLANK17  019070
      SUM=0.                                                            BLANK18  019080
      DO 20 LIMXL=1,NXX                                                 BLANK19  019090
      DO 10 JJ=1,NYY                                                    BLANK20  019100
   10 SUM=SUM+PERF(LIMXL,JJ)                                            BLANK21  019110
      IF (SUM.NE.0.) GO TO 30                                           BLANK22  019120
   20 CONTINUE                                                          BLANK23  019130
      VOID2=1                                                           BLANK24  019140
      RETURN                                                            BLANK25  019150
C                                                                       BLANK26  019160
   30 SUM=0.                                                            BLANK27  019170
      DO 50 K=LIMXL,NXX                                                 BLANK28  019180
      LIMXR=NXX-K+LIMXL                                                 BLANK29  019190
      DO 40 JJ=1,NYY                                                    BLANK30  019200
   40 SUM=SUM+PERF(LIMXR,JJ)                                            BLANK31  019210
      IF (SUM.NE.0.) GO TO 60                                           BLANK32  019220
   50 CONTINUE                                                          BLANK33  019230
C                                                                       BLANK34  019240
   60 SUM=0.                                                            BLANK35  019250
      DO 80 LIMYB=1,NYY                                                 BLANK36  019260
      DO 70 II=LIMXL,LIMXR                                              BLANK37  019270
   70 SUM=SUM+PERF(II,LIMYB)                                            BLANK38  019280
      IF (SUM.NE.0.) GO TO 90                                           BLANK39  019290
   80 CONTINUE                                                          BLANK40  019300
C                                                                       BLANK41  019310
   90 SUM=0.                                                            BLANK42  019320
      DO 110 K=LIMYB,NYY                                                BLANK43  019330
      LIMYT=NYY-K+LIMYB                                                 BLANK44  019340
      DO 100 II=LIMXL,LIMXR                                             BLANK45  019350
  100 SUM=SUM+PERF(II,LIMYT)                                            BLANK46  019360
      IF (SUM.NE.0.) RETURN                                             BLANK47  019370
  110 CONTINUE                                                          BLANK48  019380
      END                                                               BLANK49- 019390
```

```
      SUBROUTINE SUMPF                                                  *1930*11  C19400
C     THE GEOMETRY AND DAMAGE SUBROUTINES HAVE PRODUCED THE PERFORATION SUMPF 2   019410
C     ARRAY. IF THE TARGET WAS A FLAT PLATE, THE INFORMATION CONTAINED  SUMPF 3   019420
C     THEREIN IS THE NUMBER OF PERFORATIONS SUSTAINED BY EACH CELL OF A SUMPF 4   019430
C     LARGE FLAT PLATE. IF THE TARGET WAS ONE OR TWO SLANTED SIDES, THENSUMPF 5   019440
C     THIS INFORMATION IS THE TOTAL NUMBER OF PERFORATIONS SUSTAINED BY SUMPF 6   019450
C     THE SIDE(S) WHEN THE LENGTH (DIMENSION ALONG THE JUNCTION BETWEEN SUMPF 7   019460
C     THE SIDE(S) AND THE TOP) OF THE TARGET IS ONLY ONE CELL, FOR EACH SUMPF 8   019470
C     POSITION OF THIS TARGET'S CENTER. THIS SUBROUTINE SUMS THAT       SUMPF 9   019480
C     INFORMATION AND PRODUCES THE LAMBDA ARRAY WHICH CONTAINS THE TOTALSUMPF10   C19490
C     NUMBER OF PERFORATIONS SUSTAINED BY THE ENTIRE TARGET FOR EACH    SUMPF11   0195C0
C     POSITION OF THE TARGET CENTER. NOTE THAT 'TARGET', AS USED AT THISSUMPF12   019510
C     POINT, REALLY ADDRESSES A PIECE OF A COMPLETE TARGET.             SUMPF13   019520
C                                                                       SUMPF14   019530
C     FOR TARGETS WHICH EXHIBIT SYMMETRY ACROSS THE ROW BOUNDARY IN     SUMPF15   019540
C     WHICH THE PROJECTILE IS IMBEDDED, THE SYMMETRY IS UTILIZED TO     SUMPF16   019550
C     OBTAIN THE CONTRIBUTION OF THAT PART OF THE TARGET WHICH EXTENDS  SUMPF17   019560
C     INTO 'ROWS' NUMBERED LESS THAN 1.                                 SUMPF18   019570
C                                                                       SUMPF19   019580
C     II AND JJ LOCATE THE TARGET CENTER, DEFINED AS NX/2+1,NY/2+1, ON  SUMPF20   019590
C     THE GRID. III, JJJ PICK OUT GRID COLUMNS AND ROWS.                SUMPF21   019600
C                                                                       SUMPF22   019610
      COMMON /COM1/COM1(20100)                                          SUMPF23   019620
      DIMENSION PERF(100,100),LAMDA(100,100),SUM(100,100)               SUMPF24   019630
      EQUIVALENCE (COM1(1),PERF(1,1),LAMDA(1,1)),(COM1(10001),SUM(1,1)) SUMPF25   019640
C                                                                       SUMPF26   019650
      COMMON /COM3/COM3(10000)                                          SUMPF27   019660
      EQUIVALENCE  (COM3(9965),NX),(COM3(9966),NY),(COM3(9969),A)       SUMPF28   019670
     1             ,(COM3(9970),B),(COM3(9979),LIMXL),(COM3(9980),LIMXR) SUMPF29  019680
     2             ,(COM3(9981),LIMYB),(COM3(9982),LIMYT)               SUMPF30   019690
     3             ,(COM3(9998),IFLAG)                                  SUMPF31   019700
C                                                                       SUMPF32   019710
      COMMON /COLROW/ICOL,IROW,ICOL1,IROW1                              SUMPF33   019720
      COMMON /LYMIT/LYMXL,LYMXR,LYMYB,LYMYT                             SUMPF34   019730
      COMMON /NXXNYY/NXX,NYY                                            SUMPF35   019740
C                                                                       SUMPF36   019750
      INTEGER SNUM1,SNUM2,SNUM3,SNUM4,SNUM5,SNUM6                       SUMPF37   019760
      REAL LAMDA,LAMDE                                                  SUMPF38   019770
C                                                                       SUMPF39   019780
      IF (B.EQ.0..AND.NY.GT.1) GO TO 30                                 SUMPF40   019790
      LYMYB=LIMYB                                                       SUMPF41   019800
      LYMYT=LIMYT                                                       SUMPF42   019810
      IF (B.EQ.0..AND.A.NE.0.) GO TO 20                                 SUMPF43   019820
      DO 10 JJ=LYMYB,LYMYT                                              SUMPF44   019830
      DO 10 II=LIMXL,LIMXR                                              SUMPF45   019840
   10 SUM(II,JJ)=PERF(II,JJ)                                            SUMPF46   019850
      GO TO 250                                                         SUMPF47   019860
   20 LYMXL=LIMXL                                                       SUMPF48   019870
      LYMXR=LIMXR                                                       SUMPF49   019880
      RETURN                                                            SUMPF50   019890
C                                                                       SUMPF51   019900
C                                                                       SUMPF52   019910
   30 NY2=NY/2                                                          SUMPF53   019920
C     TARGET MID ROW                                                    SUMPF54   019930
      NYMID=NY2+1                                                       SUMPF55   019940
C     TARGET TOP ROW IS NYTOP ROWS ABOVE NYMID. TARGET BOTTOM ROW IS NY2SUMPF56   019950
C     ROWS BELOW NYMID                                                  SUMPF57   019960
      NYTOP=NY-NYMID                                                    SUMPF58   019970
C     BOUNDARIES FOR MOVEMENT OF TARGET MID ROW FROM BOTTOM TO TOP OF   SUMPF59   019980
C     ARRAY                                                             SUMPF60   019990
```

```
      LYMYB=LIMYB-NYTOP                                                 SUMPF61  020000
      IF (LYMYB.LT.1.AND.IROW.NE.0) IFLAG=IFLAG+4                       SUMPF62  020010
      LYMYT=LIMYT+NY2                                                   SUMPF63  020020
      IF (LYMYT.LE.NYY) GO TO 40                                        SUMPF64  020030
      IFLAG=IFLAG+8                                                     SUMPF65  020040
      LYMYT=NYY                                                         SUMPF66  020050
C     VARIOUS SUMMATION LIMITS                                          SUMPF67  020060
C     LOCATION OF TARGET TOP ROW WHEN TARGET MID ROW IS JUST BELOW ARRAYSUMPF68  020070
C     ROW 1                                                             SUMPF69  020080
   40 LIM1=NYTOP                                                        SUMPF70  020090
C     AVOID SUMMING ZEROES                                              SUMPF71  020100
      IF (LIM1.GT.LIMYT) LIM1=LIMYT                                     SUMPF72  020110
C     LOCATION OF TARGET MID ROW WHEN TARGET BOTTOM ROW IS IN LIMYB     SUMPF73  020120
      LIM2=LIMYB+NY2                                                    SUMPF74  020130
C     NO STORAGE BEYOND ARRAY LIMITS                                    SUMPF75  020140
      IF (LIM2.GT.NYY) LIM2=NYY                                         SUMPF76  020150
C     FIRST STORAGE OF LAMDE OCCURS IN THIS TARGET MID ROW POSITION     SUMPF77  020160
      LIM3=LYMYB                                                        SUMPF78  020170
C     NO STORAGE BEYOND ARRAY LIMITS                                    SUMPF79  020180
      IF (LYMYB.LT.1) LIM3=1                                            SUMPF80  020190
C     LOCATION OF TARGET MID ROW WHEN TARGET TOP ROW IS IN LIMYT        SUMPF81  020200
      LIM4=LIMYT-NYTOP                                                  SUMPF82  020210
C     IF LIM5(=LIM2).GT.LIM4, THE TARGET TOP ROW WILL BE BEYOND LIMYT   SUMPF83  020220
C     WHEN THE TARGET BOTTOM ROW IS IN LIMYB                            SUMPF84  020230
      LIM5=LIM2                                                         SUMPF85  020240
      IF (LIM5.GT.LIM4) LIM5=LIM4                                       SUMPF86  020250
      LIM6=LIM5+1                                                       SUMPF87  020260
      IF (LYMYB.LT.1.AND.LIM1.EQ.LIMYT) LIM6=1                          SUMPF88  020270
      LIM7=LIM2+1                                                       SUMPF89  020280
      IF (LIM5.LT.LIM4.AND.(LYMYB.GT.0.OR.LIM1.LT.LIMYT)) LIM7=LIM4+1   SUMPF90  020290
C     FOLLOWING 2 LIMITS ARE USED FOR SYMMETRY                          SUMPF91  020300
      LIM8=2*IROW+1                                                     SUMPF92  020310
      LIM9=2*IROW+NY2                                                   SUMPF93  020320
      IF (LIM9.GT.LIMYT) LIM9=LIMYT                                     SUMPF94  020330
      LIM10=NYMID-LIM9+2*IROW-1                                         SUMPF95  020340
C     SET UP LOGIC                                                      SUMPF96  020350
      ASSIGN 120 TO SNUM1                                               SUMPF97  020360
      IF (LYMYB.GT.0) GO TO 50                                          SUMPF98  020370
C     LYMYB.LT.1 MEANS THAT, WHEN SUMMING STARTS, NO STORAGE OF LAMDE   SUMPF99  020380
C     WILL OCCUR BECAUSE THE TARGET MID ROW WILL NOT BE WITHIN THE ARRAYSUMP100  020390
      ASSIGN 100 TO SNUM1                                               SUMP101  020400
      ASSIGN 120 TO SNUM2                                               SUMP102  020410
      IF (LIM1.LT.LIMYT) GO TO 50                                       SUMP103  020420
C     LIM1.GE.LIMYT MEANS THAT THERE ARE NO MORE NON-ZERO ARRAY ROWS TO SUMP104  020430
C     BE SUMMED                                                         SUMP105  020440
      ASSIGN 160 TO SNUM2                                               SUMP106  020450
      GO TO 60                                                          SUMP107  020460
   50 ASSIGN 140 TO SNUM3                                               SUMP108  020470
C     LIM5.LT.LIM4 MEANS THAT THE TARGET BOTTOM ROW IS IN LIMYB BEFORE  SUMP109  020480
C     THE TARGET TOP ROW REACHES LIMYT                                  SUMP110  020490
      IF (LIM5.LT.LIM4) GO TO 60                                        SUMP111  020500
      ASSIGN 160 TO SNUM3                                               SUMP112  020510
      IF (LIM5.LT.LIM2) GO TO 60                                        SUMP113  020520
C     LIM5.EQ.LIM2 MEANS THAT THE TARGET BOTTOM ROW REACHES LIMYB AS THESUMP114  020530
C     TARGET TOP ROW REACHES LIMYT                                      SUMP115  020540
      ASSIGN 180 TO SNUM3                                               SUMP116  020550
      IF (LIM7.LE.NYY) GO TO 70                                         SUMP117  020560
C     LIM7.GT.NYY MEANS THAT THERE IS NO MORE SPACE AVAILABLE FOR       SUMP118  020570
C     STORAGE, SO WE MOVE ON TO SYMMETRY                                SUMP119  020580
      ASSIGN 200 TO SNUM3                                               SUMP120  020590
```

```
C     LIM8.GT.LIMYT MEANS THAT WHAT WE WOULD PICK UP DUE TO SYMMETRY IS
C     BEYOND THE ARRAY LIMITS
      IF (LIM8.LE.LIMYT) GO TO 80
      ASSIGN 240 TO SNUM3
      GO TO 90
   60 ASSIGN 160 TO SNUM4
      IF (LIM7.LE.NYY) GO TO 70
C     LIM7.GT.NYY MEANS THAT THERE IS NO MORE SPACE AVAILABLE FOR
C     STORAGE, SO WE MOVE ON TO SYMMETRY
      ASSIGN 200 TO SNUM4
C     LIM8.GT.LIMYT MEANS THAT WHAT WE WOULD PICK UP DUE TO SYMMETRY IS
C     BEYOND THE ARRAY LIMITS
      IF (LIM8.LE.LIMYT) GO TO 80
      ASSIGN 240 TO SNUM4
      GO TO 90
   70 ASSIGN 200 TO SNUM5
C     LIM8.GT.LIMYT MEANS THAT WHAT WE WOULD PICK UP DUE TO SYMMETRY IS
C     BEYOND THE ARRAY LIMITS
      IF (LIM8.LE.LIMYT) GO TO 80
      ASSIGN 240 TO SNUM5
      GO TO 90
   80 ASSIGN 220 TO SNUM6
      IF (LIM10.LT.1) ASSIGN 240 TO SNUM6
C     DO THE SUMMATION FOR EACH CONTRIBUTING ARRAY COLUMN
   90 DO 240 II=LIMXL,LIMXR
      LAMDE=0.
      GO TO SNUM1, (100,120)
  100 DO 110 JJJ=LIMYB,LIM1
C     DO THAT PART OF THE SUMMATION FOR WHICH NO STORAGE OF LAMDE OCCURS
C     BECAUSE THE TARGET MID ROW IS BELOW ARRAY ROW 1
  110 LAMDE=LAMDE+PERF(II,JJJ)
      GO TO SNUM2, (120,160)
  120 DO 130 JJ=LIM3,LIM5
C     SUCCESSIVE ARRAY ROWS .LE.LIMYT ARE COVERED BUT NONE .GE.LIMYB ARE
C     UNCOVERED AS THE TARGET MOVES UPWARD. STORAGE OF LAMDE OCCURS.
      LAMDE=LAMDE+PERF(II,JJ+NYTOP)
  130 SUM(II,JJ)=LAMDE
      GO TO SNUM3, (140,160,180,200,240)
  140 DO 150 JJ=LIM6,LIM4
C     AS THE TARGET MOVES UPWARD, THE TARGET TOP ROW COVERS ONE AND THE
C     TARGET BOTTOM ROW UNCOVERS ANOTHER NON-ZERO ARRAY ROW. STORAGE OF
C     LAMDE OCCURS.
      LAMDE=LAMDE+PERF(II,JJ+NYTOP)-PERF(II,JJ-NYMID)
      IF (LAMDE.LT.0.) LAMDE=0.
  150 SUM(II,JJ)=LAMDE
      GO TO SNUM4, (160,200,240)
  160 DO 170 JJ=LIM6,LIM2
C     ALL NON-ZERO ARRAY ROWS REMAIN COVERED AS THE TARGET MOVES UPWARD
  170 SUM(II,JJ)=LAMDE
      GO TO SNUM4, (180,200,240)
  180 DO 190 JJ=LIM7,LYMYT
C     TARGET BOTTOM ROW UNCOVERS NON-ZERO ARRAY ROWS AS THE TARGET MOVES
C     UPWARD. TARGET TOP ROW IS BEYOND LIMYT. STORAGE OF LAMDE OCCURS.
      LAMDE=LAMDE-PERF(II,JJ-NYMID)
      IF (LAMDE.LT.0.) LAMDE=0.
  190 SUM(II,JJ)=LAMDE
      GO TO SNUM5, (200,240)
C     ALLOW FOR SYMMETRY ABOUT THE ROW BOUNDARY IN WHICH THE PROJECTILE
C     IS IMBEDDED. NOTE THAT THIS IS ALLOWED ONLY FOR A FLAT TARGET OR
C     FOR ONE OF TWO SIDES WHOSE JUNCTION WITH THE TOP IS IN THE Y
```

```
C     DIRECTION.
  200 LAMDE=0.
      DO 210 JJJ=LIM6,LIM9
      LAMDE=LAMDE+PERF(II,JJJ)
      JJ=NYMID-JJJ+2*IROW
  210 SUM(II,JJ)=SUM(II,JJ)+LAMDE
      GO TO SNUM6, (220,240)
  220 DO 230 JJ=1,LIM10
  230 SUM(II,JJ)=SUM(II,JJ)+LAMDE
  240 CONTINUE
C
      IF (LYMYB.LT.1) LYMYB=1
C
C     THIS PART OF THE SUBROUTINE IS ANALOGOUS TO THE PRECEEDING EXCEPT
C     THAT SYMMETRY IS NOT CONSIDERED
  250 IF (A.EQ.C..AND.NX.GT.1) GO TO 270
      LYMXL=LIMXL
      LYMXR=LIMXR
      DO 260 JJ=LYMYB,LYMYT
      DO 260 II=LYMXL,LYMXR
  260 LAMDA(II,JJ)=SUM(II,JJ)
      RETURN
C
C
  270 NX2=NX/2
      NXMID=NX2+1
      NXRIT=NX-NXMID
      LYMXL=LIMXL-NXRIT
      IF (LYMXL.LT.1) IFLAG=IFLAG+1
      LYMXR=LIMXR+NX2
      IF (LYMXR.LE.NXX) GO TO 280
      IFLAG=IFLAG+2
      LYMXR=NXX
  280 LIM1=NXRIT
      IF (LIM1.GT.LIMXR) LIM1=LIMXR
      LIM2=LIMXL+NX2
      IF (LIM2.GT.NXX) LIM2=NXX
      LIM3=LYMXL
      IF (LYMXL.LT.1) LIM3=1
      LIM4=LIMXR-NXRIT
      LIM5=LIM2
      IF (LIM5.GT.LIM4) LIM5=LIM4
      LIM6=LIM5+1
      IF (LYMXL.LT.1.AND.LIM1.EQ.LIMXR) LIM6=1
      LIM7=LIM2+1
      IF (LIM5.LT.LIM4.AND.(LYMXL.GT.C.OR.LIM1.LT.LIMXR)) LIM7=LIM4+1
      ASSIGN 340 TO SNUM1
      IF (LYMXL.GT.C) GO TO 290
      ASSIGN 320 TO SNUM1
      ASSIGN 340 TO SNUM2
      IF (LIM1.LT.LIMXR) GO TO 290
      ASSIGN 380 TO SNUM2
      GO TO 300
  290 ASSIGN 360 TO SNUM3
      IF (LIM5.LT.LIM4) GO TO 300
      ASSIGN 380 TO SNUM3
      IF (LIM5.LT.LIM2) GO TO 300
      ASSIGN 400 TO SNUM3
      IF (LIM7.GT.NXX) ASSIGN 420 TO SNUM3
      GO TO 310
```

```
300 ASSIGN 400 TO SNUM4                                          SUMP241   021800
    IF (LIM7.GT.NXX) ASSIGN 420 TO SNUM4                         SUMP242   021810
310 DO 420 JJ=LYMYB,LYMYT                                        SUMP243   021820
    LAMDE=0.                                                     SUMP244   021830
    GO TO SNUM1, (320,340)                                       SUMP245   021840
320 DO 330 III=LIMXL,LIM1                                        SUMP246   021850
330 LAMDE=LAMDE+SUM(III,JJ)                                      SUMP247   021860
    GO TO SNUM2, (340,380)                                       SUMP248   021870
340 DO 350 II=LIM3,LIM5                                          SUMP249   021880
    LAMDE=LAMDE+SUM(II+NXRIT,JJ)                                 SUMP250   021890
350 LAMDA(II,JJ)=LAMDE                                           SUMP251   021900
    GO TO SNUM3, (360,380,400,420)                               SUMP252   021910
360 DO 370 II=LIM6,LIM4                                          SUMP253   021920
    LAMDE=LAMDE+SUM(II+NXRIT,JJ)-SUM(II-NXMID,JJ)                SUMP254   021930
    IF (LAMDE.LT.0.) LAMDE=0.                                    SUMP255   021940
370 LAMDA(II,JJ)=LAMDE                                           SUMP256   021950
    GO TO SNUM4, (400,420)                                       SUMP257   021960
380 DO 390 II=LIM6,LIM2                                          SUMP258   021970
390 LAMDA(II,JJ)=LAMDE                                           SUMP259   021980
    GO TO SNUM4, (400,420)                                       SUMP260   021990
400 DO 410 II=LIM7,LYMXR                                         SUMP261   022000
    LAMDE=LAMDE-SUM(II-NXMID,JJ)                                 SUMP262   022010
    IF (LAMDE.LT.0.) LAMDE=0.                                    SUMP263   022020
410 LAMDA(II,JJ)=LAMDE                                           SUMP264   022030
420 CONTINUE                                                     SUMP265   022040
    IF (LYMXL.LT.1) LYMXL=1                                      SUMP266   022050
    RETURN                                                       SUMP267   022060
    END                                                          SUMP268-  022070
```

```
      SUBROUTINE VOID
C     THIS SUBROUTINE IS CALLED IF THERE ARE NO PERFORATIONS. IT
C     GENERATES A SMALL, NULL LAMDA ARRAY FOR THE TAPE WRITING WHICH
C     OCCURS LATER.
C
      COMMON /COM1/COM1(20100)
      DIMENSION LAMDA(100,100)
      EQUIVALENCE (COM1(1),LAMDA(1,1))
C
      COMMON /COM3/COM3(10000)
      EQUIVALENCE  (COM3(9979),LIMXL),(COM3(9980),LIMXR)
     1             ,(COM3(9981),LIMYB),(COM3(9982),LIMYT)
C
      COMMON /COLROW/ICOL,IROW,ICOL1,IROW1
      COMMON /LYMIT/LYMXL,LYMXR,LYMYB,LYMYT
      COMMON /NXXNYY/NXX,NYY
C
      REAL LAMDA
C
      LYMXL=ICOL
      LYMYB=IROW
      IF (LYMXL.LT.1) LYMXL=1
      IF (LYMXL.GT.NXX-1) LYMXL=NXX-1
      IF (LYMYB.LT.1) LYMYB=1
      IF (LYMYB.GT.NYY-1) LYMYB=NYY-1
      LYMXR=LYMXL+1
      LYMYT=LYMYB+1
      LIMYB=LYMYB
      LIMYT=LYMYT
      LIMXL=LYMXL
      LIMXR=LYMXR
      LAMDA(LYMXL,LYMYB)=0.
      LAMDA(LYMXL,LYMYT)=0.
      LAMDA(LYMXR,LYMYB)=0.
      LAMDA(LYMXR,LYMYT)=0.
      RETURN
      END
```

```
      SUBROUTINE PAIR
C     THIS SUBROUTINE PIECES TOGETHER 2 EXPECTED PERFORATIONS MATRICES
C     (EACH REPRESENTING A HALF-INFINITE PLANE) TO OBTAIN A COMPLETE
C     MATRIX
C
      COMMON /COM1/COM1(20100)
      DIMENSION LAMDA2(100,100),LAMDA1(100,100)
      EQUIVALENCE (COM1(1),LAMDA2(1,1)),(COM1(10001),LAMDA1(1,1))
C
      COMMON /COM3/COM3(10000)
      DIMENSION FILE2(7),SHFTX2(7),SHFTY2(2)
      EQUIVALENCE  (COM3(8439),FILE2(1)),(COM3(8446),SHFTX2(1))
     1             ,(COM3(6453),SHFTY2(1))
C
      COMMON /DISC/DISC(15016)
      EQUIVALENCE (DISC(15010),NFILE),(DISC(15012),PTAPE1)
C
      COMMON /CELL/CELL
      COMMON /COLROW/ICOL,IROW,ICOL1,IROW1
      COMMON /CONSTT/CONSTT(6)
      EQUIVALENCE (CONSTT(5),DIMX),(CONSTT(6),DIMY)
      COMMON /HOB/IHOB,NHOB,HOB
      COMMON /KSYM/KSYM
      COMMON /LQMIT/LQMXL,LQMXR
      COMMON /LYMIT/LYMXL,LYMXR,LYMYB,LYMYT
      COMMON /SHIFTS/FILE(7),SHFTX(7),SHFTY(7)
C
      EQUIVALENCE (SLIP1,SLIP2,ITEMP),(SLIP3,SLIP4)
C
      INTEGER DIMX,DIMY,FILE,FILE2,PTAPE1,SHFTX,SHFTY,SHFTX2,SHFTY2
     1        ,SLIPX1,SLIPX2,SLIPY2,SLIP1,SLIP2,SLIP3,SLIP4
      REAL LAMDA1,LAMDA2
C
C     REFLECT THE SECOND HALF OF THE PAIR THROUGH THE PROJECTILE
C     TRAJECTORY
      NFILE2=NFILE
      DO 10 I=1,NFILE2
      FILE2(I)=FILE(I)
      SHFTX2(I)=SHFTX(I)
   10 SHFTY2(I)=SHFTY(I)
C
      SLIPX2=ICOL-ICOL1
      SLIPX1=0
      SLIP4=DIMX-LYMXR
      IF (SLIP4.LT.0) SLIP4=0
      IF (SLIPX2) 30,50,20
C
   20 IF (SLIPX2.LE.SLIP4) GO TO 40
      SLIP1=1-LQMXL
      IF (SLIP1.GT.0) SLIP1=0
      SLIPX1=SLIP4-SLIPX2
      SLIPX2=SLIP4
      IF (SLIPX1.GE.SLIP1) GO TO 40
      SLIPX1=SLIP1
      SLIPX2=ICOL-ICOL1+SLIP1
      GO TO 40
C
   30 SLIP2=1-LYMXL
      IF (SLIP2.GT.0) SLIP2=0
      IF (SLIPX2.GE.SLIP2) GO TO 40
```

```
      SLIP3=DIMX-LQMXR                                                  PAIR 61   023050
      IF (SLIP3.LT.0) SLIP3=0                                           PAIR 62   023060
      SLIPX1=SLIP2-SLIPX2                                               PAIR 63   023070
      SLIPX2=SLIP2                                                      PAIR 64   023080
      IF (SLIPX1.LE.SLIP3) GO TO 40                                     PAIR 65   023090
      SLIPX1=SLIP3                                                      PAIR 66   023100
      SLIPX2=ICOL-ICOL1+SLIP3                                           PAIR 67   023110
C     AT THIS POINT WE HAVE SLIPX1 AND SLIPX2                           PAIR 68   023120
C     SLIPX2 IS THE AMOUNT BY WHICH THE CURRENT ARRAY (IN LAMDA1) IS    PAIR 69   023130
C     SHIFTED AS IT MOVES INTO LAMDA2. PART OF THIS ARRAY MAY BE CHOPPEDPAIR 70   023140
C     OFF. SLIPX1 IS THE AMOUNT BY WHICH THE ARRAY ON TAPE WILL BE      PAIR 71   023150
C     SHIFTED WITHIN LAMDA1 AFTER IT IS READ IN.                        PAIR 72   023160
   40 LYMXL=LYMXL+SLIPX2                                                PAIR 73   023170
      LYMXR=LYMXR+SLIPX2                                                PAIR 74   023180
      IF (LYMXL.LT.1) LYMXL=1                                           PAIR 75   023190
      IF (LYMXR.GT.DIMX) LYMXR=DIMX                                     PAIR 76   023200
   50 SLIPY2=DIMY+1+IROW1                                               PAIR 77   023210
      ITEMP=LYMYB                                                       PAIR 78   023220
      LYMYB=SLIPY2-LYMYT                                                PAIR 79   023230
      LYMYT=SLIPY2-ITEMP                                                PAIR 80   023240
C     DOESN'T MATTER IF WE SPILL OUT THE TOP HERE                       PAIR 81   023250
      DO 60 I=LYMXL,LYMXR                                               PAIR 82   023260
      DO 60 J=LYMYB,LYMYT                                               PAIR 83   023270
   60 LAMDA2(I,J)=LAMDA1(I-SLIPX2,SLIPY2-J)                             PAIR 84   023280
C     LOW ORDER ROWS OF THE 100X200 ARRAY ARE IN PLACE WITH X AND Y     PAIR 85   023290
C     LIMITS LYMXL, LYMXR, LYMYB, LYMYT                                 PAIR 86   023300
C     READ IN THE HIGH ORDER ROWS WITH NO SHIFTING                      PAIR 87   023310
      READ (PTAPE1) ICOL,IROW,LQMXL,LQMXR,LQMYB,LQMYT,NFILE,((LAMDA1(I, PAIR 88R  023320
     1 J),I=LQMXL,LQMXR),J=LQMYB,LQMYT),(FILE(I),SHFTX(I),SHFTY(I),I=1,PAIR 89R  023330
     2 NFILE)                                                           PAIR 90R  023340
      BACKSPACE PTAPE1                                                  PAIR 91T  023350
      IF (LQMYB.LE.IROW) LQMYB=IROW+1                                   PAIR 92   023360
C     MUST START WITH THE SIDE TOWARD WHICH WE SHIFT                    PAIR 93   023370
      IF (SLIPX1.GT.0) GO TO 80                                         PAIR 94   023380
      DO 70 I=LQMXL,LQMXR                                               PAIR 95   023390
      DO 70 J=LQMYB,LQMYT                                               PAIR 96   023400
   70 LAMDA1(I+SLIPX1,J-IROW)=LAMDA1(I,J)                               PAIR 97   023410
      GO TO 100                                                         PAIR 98   023420
   80 DO 90 I=LQMXL,LQMXR                                               PAIR 99   023430
      II=LQMXR+LQMXL-I                                                  PAIR100   023440
      DO 90 J=LQMYB,LQMYT                                               PAIR101   023450
   90 LAMDA1(II+SLIPX1,J-IROW)=LAMDA1(II,J)                             PAIR102   023460
  100 LQMXL=LQMXL+SLIPX1                                                PAIR103   023470
      LQMXR=LQMXR+SLIPX1                                                PAIR104   023480
      LQMYB=LQMYB-IROW                                                  PAIR105   023490
      LQMYT=LQMYT-IROW                                                  PAIR106   023500
C     BOTH PARTS OF THE 100X200 ARRAY ARE IN. STRAIGHTEN THE SIDES WITH PAIR107   023510
C     ZEROES.                                                           PAIR108   023520
      IF (LQMXL-LYMXL) 130,150,110                                      PAIR109   023530
  110 ITEMP=LQMXL-1                                                     PAIR110   023540
      DO 120 J=LQMYB,LQMYT                                              PAIR111   023550
      DO 120 I=LYMXL,ITEMP                                              PAIR112   023560
  120 LAMDA1(I,J)=0.                                                    PAIR113   023570
      GO TO 150                                                         PAIR114   023580
  130 ITEMP=LYMXL-1                                                     PAIR115   023590
      DO 140 J=LYMYB,LYMYT                                              PAIR116   023600
      DO 140 I=LQMXL,ITEMP                                              PAIR117   023610
  140 LAMDA2(I,J)=0.                                                    PAIR118   023620
      LYMXL=LQMXL                                                       PAIR119   023630
C     LOW X SIDE IS STRAIGHT                                            PAIR120   023640
```

```
  150 IF (LQMXR-LYMXR) 160,200,180                                    PAIR121  023650
  160 ITEMP=LQMXR+1                                                   PAIR122  023660
      DO 170 J=LQMYB,LQMYT                                            PAIR123  023670
      DO 170 I=ITEMP,LYMXR                                            PAIR124  023680
  170 LAMDA1(I,J)=0.                                                  PAIR125  023690
      GO TO 200                                                       PAIR126  023700
  180 ITEMP=LYMXR+1                                                   PAIR127  023710
      DO 190 J=LYMYB,LYMYT                                            PAIR128  023720
      DO 190 I=ITEMP,LQMXR                                            PAIR129  023730
  190 LAMDA2(I,J)=0.                                                  PAIR130  023740
      LYMXR=LQMXR                                                     PAIR131  023750
C     NOW WE NEED ONLY INSURE THERE IS NO GAP IN THE MIDDLE           PAIR132  023760
  200 IF (LQMYB+DIMY.EQ.LYMYT+1) GO TO 220                            PAIR133  023770
      LYMYT=LYMYT+1                                                   PAIR134  023780
      LQMYB=LQMYB+DIMY-1                                              PAIR135  023790
      DO 210 I=LYMXL,LYMXR                                            PAIR136  023800
      DO 210 J=LYMYT,LQMYB                                            PAIR137  023810
  210 LAMDA2(I,J)=0.                                                  PAIR138  023820
  220 LYMYT=LQMYT+DIMY                                                PAIR139  023830
C     DONE. LIMITS ARE LYMXL, LYMXR, LYMYB, LYMYT                     PAIR140  023840
      ICOL1=ICOL+SLIPX1                                               PAIR141  023850
      IROW1=DIMY                                                      PAIR142  023860
      WRITE (PTAPE1) HOB,CELL,KSYM,ICOL1,IROW1,LYMXL,LYMXR,LYMYB,LYMYT, PAIR143W 023870
     1  NFILE,((LAMDA2(I,J),I=LYMXL,LYMXR),J=LYMYB,LYMYT),(FILE(I),    PAIR144W 023880
     2  SHFTX(I),SHFTY(I),I=1,NFILE)                                  PAIR145W 023890
      WRITE (PTAPE1) NFILE2,(FILE2(I),SHFTX2(I),SHFTY2(I),I=1,NFILE2)  PAIR146W 023900
      RETURN                                                          PAIR147  023910
      END                                                             PAIR148- 023920
```

```
      SUBROUTINE PEKAY (LAP1)                                             *2363*14  023930
C     THIS SUBROUTINE GENERATES THE PROBABILITY OF DAMAGE MATRIX AND      PEKAY 2   023940
C     COMPUTES THE LETHAL AREA, THE AREA IN WHICH PD IS NON ZERO          PEKAY 3   023950
C     (GREATER THAN .0005), AND THE AVERAGE PD IN THAT AREA.              PEKAY 4   023960
C                                                                         PEKAY 5   023970
      COMMON /COM1/COM1(20100)                                            PEKAY 6   023980
      DIMENSION LAMDA(100,200),PK(100,200)                                PEKAY 7   023990
      EQUIVALENCE (COM1(1),LAMDA(1,1),PK(1,1))                            PEKAY 8   024000
C                                                                         PEKAY 9   024010
      COMMON /COM3/COM3(10000)                                            PEKAY10   024020
      DIMENSION KILL(25),DKILL(25),LETHL(25),DLETHL(25)                   PEKAY11   024030
      EQUIVALENCE  (COM3(4420),KILL(1)),(COM3(4445),DKILL(1))             PEKAY12   024040
     1             ,(COM3(4470),LETHL(1)),(COM3(4495),DLETHL(1))          PEKAY13   024050
     2             ,(COM3(9975),IKRIT),(COM3(9976),KRYT)                  PEKAY14   024060
     3             ,(COM3(10000),PKAY)                                    PEKAY15   024070
C                                                                         PEKAY16   024080
      COMMON /CELL/CELL                                                   PEKAY17   024090
      COMMON /COLROW/ICOL,IROW,ICOL1,IROW1                                PEKAY18   024100
      COMMON /CONSTT/CONSTT(6)                                            PEKAY19   024110
      EQUIVALENCE (CONSTT(4),FTTOM)                                       PEKAY20   024120
      COMMON /HOB/IHOB,NHOB,HOB                                           PEKAY21   024130
      EQUIVALENCE (IHOB,IPNT)                                             PEKAY22   024140
      COMMON /KRIT/KRIT(7)                                                PEKAY23   024150
      COMMON /KSYM/KSYM                                                   PEKAY24   024160
      COMMON /LIMIT/LIMIT1,LIMIT2,LIMIT3,LIMIT4                           PEKAY25   024170
      COMMON /LYMIT/LYMXL,LYMXR,LYMYB,LYMYT                               PEKAY26   024180
      COMMON /NXXNYY/NXX,NYY                                              PEKAY27   024190
C                                                                         PEKAY28   024200
      REAL LAMDA,KILL,LETHL                                               PEKAY29   024210
C                                                                         PEKAY30   024220
      ASSIGN 10 TO JUMP1                                                  PEKAY31   024230
      ASSIGN 110 TO JUMP2                                                 PEKAY32   024240
      AREA=0.                                                             PEKAY33   024250
      ISUM=0                                                              PEKAY34   024260
      DLETHL(IPNT)=0.                                                     PEKAY35   024270
      DKILL(IPNT)=0.                                                      PEKAY36   024280
      JTEMP1=LYMYB                                                        PEKAY37   024290
      JTEMP2=LYMYT                                                        PEKAY38   024300
      IF (KSYM.EQ.2.OR.LYMYB.GT.IROW1.OR.LYMYT.LE.IROW1) GO TO 120        PEKAY39   024310
      JTEMP2=IROW1                                                        PEKAY40   024320
   10 IF (KRIT(IKRIT).EQ.1) GO TO 20                                      PEKAY41   024330
      ASSIGN 280 TO JUMP1                                                 PEKAY42   024340
      GO TO 270                                                           PEKAY43   024350
C     LOOP ON THE TARGET CENTER POSITIONS                                 PEKAY44   024360
   20 DO 90 JJ=JTEMP1,JTEMP2                                              PEKAY45   024370
      DO 90 II=LYMXL,LYMXR                                                PEKAY46   024380
      PEK=LAMDA(II,JJ)                                                    PEKAY47   024390
      IF (PEK-.000500012500) 30,60,40                                     PEKAY48   024400
C     1.-EXP(-X)>.0005 IF X>.000500012500                                 PEKAY49   024410
   30 PK(II,JJ)=0.                                                        PEKAY50   024420
      GO TO 90                                                            PEKAY51   024430
   40 IF (PEK-11.51292546) 70,50,50                                       PEKAY52   024440
C     1.-EXP(-X)>.99999 IF X>11.51292546                                  PEKAY53   024450
   50 PEK=1.                                                              PEKAY54   024460
      GO TO 80                                                            PEKAY55   024470
   60 PEK=.0005                                                           PEKAY56   024480
      GO TO 80                                                            PEKAY57   024490
   70 PEK=1.-EXP(-PEK)                                                    PEKAY58   024500
   80 ISUM=ISUM+1                                                         PEKAY59   024510
      AREA=AREA+PEK                                                       PEKAY60   024520
```

```
      PK(II,JJ)=PEK                                                     PEKAY61   024530
   90 CONTINUE                                                          PEKAY62   024540
      ASSIGN 20 TO JUMP1                                                PEKAY63   024550
  100 GO TO JUMP2, (110,130)                                            PEKAY64   024560
  110 DLETHL(IPNT)=AREA*(CELL*FTTOM)**2                                 PEKAY65   024570
      DKILL(IPNT)=FLOAT(ISUM)*(CELL*FTTOM)**2                           PEKAY66   024580
      JTEMP1=JTEMP2+1                                                   PEKAY67   024590
      JTEMP2=LYMYT                                                      PEKAY68   024600
  120 ASSIGN 130 TO JUMP2                                               PEKAY69   024610
      GO TO JUMP1, (10,20,280)                                          PEKAY70   024620
  130 FSUM=ISUM                                                         PEKAY71   024630
      LETHL(IPNT)=AREA*(CELL*FTTOM)**2                                  PEKAY72   024640
      KILL(IPNT)=FSUM*(CELL*FTTOM)**2                                   PEKAY73   024650
      PKAY=0.                                                           PEKAY74   024660
      IF (ISUM.NE.0) PKAY=AREA/FSUM                                     PEKAY75   024670
      LAP1=ISUM                                                         PEKAY76   024680
      IF (ISUM.EQ.0) GO TO 260                                          PEKAY77   024690
      SUM=0.                                                            PEKAY78   024700
      DO 150 LIMIT=LYMYB,LYMYT                                          PEKAY79   024710
      LIMIT4=LYMYT+LYMYB-LIMIT                                          PEKAY80   024720
      DO 140 I=LYMXL,LYMXR                                              PEKAY81   024730
  140 SUM=SUM+PK(I,LIMIT4)                                              PEKAY82   024740
      IF (SUM.NE.0.) GO TO 160                                          PEKAY83   024750
  150 CONTINUE                                                          PEKAY84   024760
  160 SUM=0.                                                            PEKAY85   024770
      DO 180 LIMIT=LYMXL,LYMXR                                          PEKAY86   024780
      LIMIT2=LYMXR+LYMXL-LIMIT                                          PEKAY87   024790
      DO 170 J=LYMYB,LIMIT4                                             PEKAY88   024800
  170 SUM=SUM+PK(LIMIT2,J)                                              PEKAY89   024810
      IF (SUM.NE.0.) GO TO 190                                          PEKAY90   024820
  180 CONTINUE                                                          PEKAY91   024830
  190 SUM=0.                                                            PEKAY92   024840
      DO 210 LIMIT1=LYMXL,LYMXR                                         PEKAY93   024850
      DO 200 J=LYMYB,LIMIT4                                             PEKAY94   024860
  200 SUM=SUM+PK(LIMIT1,J)                                              PEKAY95   024870
      IF (SUM.NE.0.) GO TO 220                                          PEKAY96   024880
  210 CONTINUE                                                          PEKAY97   024890
  220 SUM=0.                                                            PEKAY98   024900
      DO 240 LIMIT3=LYMYB,LYMYT                                         PEKAY99   024910
      DO 230 I=LIMIT1,LIMIT2                                            PEKA100   024920
  230 SUM=SUM+PK(I,LIMIT3)                                              PEKA101   024930
      IF (SUM.NE.0.) GO TO 250                                          PEKA102   024940
  240 CONTINUE                                                          PEKA103   024950
  250 RETURN                                                            PEKA104   024960
C                                                                       PEKA105   024970
  260 LIMIT1=ICOL1                                                      PEKA106   024980
      LIMIT3=IROW1                                                      PEKA107   024990
      IF (LIMIT1.LT.1) LIMIT1=1                                         PEKA108   025000
      IF (LIMIT3.LT.1) LIMIT3=1                                         PEKA109   025010
      IF (LIMIT1.GT.NXX-1) LIMIT1=NXX-1                                 PEKA110   025020
      II=NYY                                                            PEKA111   025030
      IF (KSYM.EQ.2) II=2*NYY                                           PEKA112   025040
      IF (LIMIT3.GT.II-1) LIMIT3=II-1                                   PEKA113   025050
      LIMIT2=LIMIT1+1                                                   PEKA114   025060
      LIMIT4=LIMIT3+1                                                   PEKA115   025070
      GO TO 250                                                         PEKA116   025080
C                                                                       PEKA117   025090
  270 KRYT=KRIT(IKRIT)-1                                                PEKA118   025100
C     LOOP ON THE TARGET CENTER POSITIONS                               PEKA119   025110
  280 DO 320 JJ=JTEMP1,JTEMP2                                           PEKA120   025120
```

```
      DO 320 II=LYMXL,LYMXR                                          PEKA121  025130
      X=LAMDA(II,JJ)                                                 PEKA122  025140
      IF (X.LT.675.84) GO TO 290                                     PEKA123  025150
      PEK=1.                                                         PEKA124  025160
      GO TO 310                                                      PEKA125  025170
  290 SUM=1.                                                         PEKA126  025180
      TERM=1.                                                        PEKA127  025190
      DO 300 J=1,KRYT                                                PEKA128  025200
      FJ=J                                                           PEKA129  025210
      TERM=TERM*X/FJ                                                 PEKA130  025220
  300 SUM=SUM+TERM                                                   PEKA131  025230
      PEK=1.-EXP(-X)*SUM                                             PEKA132  025240
      IF (PEK.GE..0005) GO TO 310                                    PEKA133  025250
      PEK=0.                                                         PEKA134  025260
      GO TO 320                                                      PEKA135  025270
  310 ISUM=ISUM+1                                                    PEKA136  025280
      AREA=AREA+PEK                                                  PEKA137  025290
  320 PK(II,JJ)=PEK                                                  PEKA138  025300
      GO TO 100                                                      PEKA139  025310
      END                                                            PEKA140- 025320
```

```
      SUBROUTINE OUTPUT (LAP1)                                          *2523*15 025330
C     THIS SUBROUTINE OUTPUTS THE PD MATRIX, THE LETHAL AREA, THE AREA  OUTPT 2  025340
C     IN WHICH PD IS NON ZERO (DAMAGE AREA), AND THE AVERAGE PD IN THAT OUTPT 3  025350
C     AREA.                                                             OUTPT 4  025360
C                                                                       OUTPT 5  025370
      COMMON /COM1/COM1(20100)                                          OUTPT 6  025380
      DIMENSION PK(100,200)                                             OUTPT 7  025390
      EQUIVALENCE (COM1(1),PK(1,1))                                     OUTPT 8  025400
C                                                                       OUTPT 9  025410
      COMMON /COM3/COM3(10000)                                          OUTPT10  025420
      DIMENSION KILL(25),DKILL(25),LETHL(25),DLETHL(25)                 OUTPT11  025430
      EQUIVALENCE  (COM3(4420),KILL(1))                                 OUTPT12  025440
     1            ,(COM3(4445),DKILL(1)),(COM3(4470),LETHL(1))          OUTPT13  025450
     2            ,(COM3(4495),DLETHL(1)),(COM3(10000),PKAY)            OUTPT14  025460
C                                                                       OUTPT15  025470
      COMMON /COLROW/ICOL,IROW,ICOL1,IROW1                              OUTPT16  025480
      COMMON /HOB/IHOB,NHOB,HOB                                         OUTPT17  025490
      EQUIVALENCE (IHOB,IPNT)                                           OUTPT18  025500
      COMMON /KSYM/KSYM                                                 OUTPT19  025510
      COMMON /LIMIT/LIMIT1,LIMIT2,LIMIT3,LIMIT4                         OUTPT20  025520
      EQUIVALENCE (LIMIT1,LYMXL),(LIMIT2,LYMXR),(LIMIT3,LYMYB)          OUTPT21  025530
     1            ,(LIMIT4,LYMYT)                                       OUTPT22  025540
C     NOTE THAT THE 4 LYM-- VARIABLES ARE NOT THE ONES STORED IN COMMON OUTPT23  025550
      COMMON /NOPK/NOPK                                                 OUTPT24  025560
      COMMON /PAGE/LINES,PAGE                                           OUTPT25  025570
      COMMON /VOID2/VOID2                                               OUTPT26  025580
C                                                                       OUTPT27  025590
      DIMENSION IFMT1(4),IFMT2(5)                                       OUTPT28  025600
C                                                                       OUTPT29  025610
      INTEGER PAGE,USED,VOID2                                           OUTPT30  025620
      REAL KILL,LETHL                                                   OUTPT31  025630
C                                                                       OUTPT32  025640
      DATA (IFMT1(I),I=1,4)/10H(2H  ,I3,2, 2HX,         ,10H20F6.3,I5)  OUTPT33D 025650
     1       ,10H20F6.3,I5)/                                            OUTPT34D 025660
      DATA (IFMT2(I),I=1,5)/10H(/1H ,5X ,,10H    20I6 ),10H    20I6 )   OUTPT35D 025670
     1       ,10H(/1H ,5X ,,10H(1H1/  6X,/                              OUTPT36D 025680
C                                                                       OUTPT37  025690
      IF (VOID2.NE.0) GO TO 90                                          OUTPT38  025700
      WRITE (6,100) LETHL(IPNT),PKAY,KILL(IPNT)                         OUTPT39W 025710
      LINES=LINES+2                                                     OUTPT40  025720
      IF (KSYM.EQ.2.OR.LYMYB.GT.IROW1.OR.LYMYT.LE.IROW1) GO TO 10       OUTPT41  025730
      F1=LETHL(IPNT)-DLETHL(IPNT)                                       OUTPT42  025740
      F2=KILL(IPNT)-DKILL(IPNT)                                         OUTPT43  025750
      PKAY=0.                                                           OUTPT44  025760
      IF (ABS(F2).GT..00001) PKAY=F1/F2                                 OUTPT45  025770
      WRITE (6,120) F1,PKAY,F2                                          OUTPT46W 025780
      LINES=LINES+4                                                     OUTPT47  025790
   10 IF (NOPK.NE.10HNO PROBABI.AND.LAP1.NE.0) GO TO 20                 OUTPT48  025800
      WRITE (6,110)                                                     OUTPT49W 025810
      LINES=LINES+7                                                     OUTPT50  025820
      GO TO 80                                                          OUTPT51  025830
   20 KNT=LYMYT-LYMYB+1                                                 OUTPT52  025840
      KNT1=(LYMXR-LYMXL+1)/20                                           OUTPT53  025850
      KEY=LYMXR-LYMXL+1-20*KNT1                                         OUTPT54  025860
      IF (KNT1.EQ.0) GO TO 60                                           OUTPT55  025870
      DO 50 I=1,KNT1                                                    OUTPT56  025880
      IF (I.EQ.1) GO TO 30                                              OUTPT57  025890
      LINES=USED(LINES)                                                 OUTPT58  025900
      IF (PAGE-LINES.LE.KNT+5) GO TO 30                                 OUTPT59  025910
      IFMT2(1)=IFMT2(5)                                                 OUTPT60  025920
```

```
      LINES=-1                                                          OUTPT61  025930
   30 ISTRT=LYMXL+(I-1)*20                                              OUTPT62  025940
      ISTOP=ISTRT+19                                                    OUTPT63  025950
      WRITE (6,IFMT2) (IICOL,IICOL=ISTRT,ISTOP)                         OUTPT64W 025960
      IFMT2(1)=IFMT2(4)                                                 OUTPT65  025970
      WRITE (6,130)                                                     OUTPT66W 025980
      DO 40 J=1,KNT                                                     OUTPT67  025990
      IIROW=LYMYT+1-J                                                   OUTPT68  026000
      WRITE (6,IFMT1) IIROW,(PK(IICOL,IIROW),IICOL=ISTRT,ISTOP),IIROW   OUTPT69W 026010
   40 CONTINUE                                                          OUTPT70  026020
      WRITE (6,IFMT2) (IICOL,IICOL=ISTRT,ISTOP)                         OUTPT71W 026030
      LINES=LINES+KNT+5                                                 OUTPT72  026040
   50 CONTINUE                                                          OUTPT73  026050
      IF (KEY.EQ.0) RETURN                                              OUTPT74  026060
      LINES=USED(LINES)                                                 OUTPT75  026070
      IF (PAGE-LINES.GE.KNT+5) GO TO 60                                 OUTPT76  026080
      IFMT2(1)=IFMT2(5)                                                 OUTPT77  026090
      LINES=-1                                                          OUTPT78  026100
   60 ISTRT=LYMXL+KNT1*20                                               OUTPT79  026110
      ENCODE (10,140,IFMT1(3) )KEY                                      OUTPT80* 026120
      ENCODE (10,150,IFMT2(2) )KEY                                      OUTPT81* 026130
      WRITE (6,IFMT2) (IICOL,IICOL=ISTRT,LYMXR)                         OUTPT82W 026140
      IFMT2(1)=IFMT2(4)                                                 OUTPT83  026150
      WRITE (6,130)                                                     OUTPT84W 026160
      DO 70 J=1,KNT                                                     OUTPT85  026170
      IIROW=LYMYT+1-J                                                   OUTPT86  026180
      WRITE (6,IFMT1) IIROW,(PK(IICOL,IIROW),IICOL=ISTRT,LYMXR),IIROW   OUTPT87W 026190
   70 CONTINUE                                                          OUTPT88  026200
      WRITE (6,IFMT2) (IICOL,IICOL=ISTRT,LYMXR)                         OUTPT89W 026210
      LINES=LINES+KNT+5                                                 OUTPT90  026220
      IFMT1(3)=IFMT1(4)                                                 OUTPT91  026230
      IFMT2(2)=IFMT2(3)                                                 OUTPT92  026240
   80 RETURN                                                            OUTPT93  026250
   90 WRITE (6,160)                                                     OUTPT94W 026260
      LINES=LINES+7                                                     OUTPT95  026270
      IF (VOID2.NE.2) GO TO 80                                          OUTPT96  026280
      WRITE (6,170)                                                     OUTPT97W 026290
      LINES=LINES+1                                                     OUTPT98  026300
      GO TO 80                                                          OUTPT99  026310
C                                                                       OUTPT100 026320
  100 FORMAT (13H LETHAL AREA=,F7.3,14H SQUARE METERS,4X,31HAVERAGE PROBOUTPT101 026330
     1ABILITY OF DAMAGE =,F5.3,6H OVER ,F6.3,14H SQUARE METERS/)        OUTPT102 026340
  110 FORMAT (//////1H ,35X,62HOUTPUT OF THE PROBABILITY OF DAMAGE MATRIOUTPT103 026350
     1X HAS BEEN SUPPRESSED)                                            OUTPT104 026360
  120 FORMAT (99H CORRESPONDING VALUES FOR ONLY THE ROWS OF HIGHER ORDEROUTPT105 026370
     1 THAN THE PROJECTILE LOCATION ARE AS FOLLOWS//14H LETHAL AREA =,F6OUTPT106 026380
     2.3,14H SQUARE METERS,4X,31HAVERAGE PROBABILITY OF DAMAGE =,F5.3,6HOUTPT107 026390
     3 OVER ,F6.3,14H SQUARE METERS/)                                   OUTPT108 026400
  130 FORMAT (1H )                                                      OUTPT109 026410
  140 FORMAT (I2,8HF6.3,I5))                                            OUTPT110 026420
  150 FORMAT (I6,4HI6 ))                                                OUTPT111 026430
  160 FORMAT (//////64H THE LETHAL AREA IS ZERO BECAUSE NO PERFORATIONS WOUTPT112 026440
     1ERE SUSTAINED./)                                                  OUTPT113 026450
  170 FORMAT (42H FURTHERMORE, THE PRESENTED AREA WAS ZERO.)            OUTPT114 026460
      END                                                               OUTP115- 026470
```

```
      SUBROUTINE CONTR                                                  *2638*16  026480
C                                                                       CONTR 2   026490
      COMMON /COM1/COM1(20100)                                          CONTR 3   026500
      DIMENSION PK(100,200),PPKK(20000),KONTR(100,200),KUNTR(20000)     CONTR 4   026510
      EQUIVALENCE (COM1(1),PK(1,1),PPKK(1),KONTR(1,1),KUNTR(1))         CONTR 5   026520
C                                                                       CONTR 6   026530
      COMMON /COM3/COM3(10000)                                          CONTR 7   026540
      DIMENSION PKBAR(13),RANGEM(13),DEFM(13),MIDRGM(13),MIDDFM(13)     CONTR 8   026550
     1         ,TEMPLA(12),TEMPKA(12),FEMPKA(12),RANGEC(13),DEFC(13)    CONTR 9   026560
     2         ,MIDRGC(13),MIDDFC(13),BOUND(13,4),BUFFER(97),ABUFFR(97) CONTR10   026570
     3         ,HEADER(124)                                             CONTR11   026580
      EQUIVALENCE  (COM3(4520),PKBAR(1))                                CONTR12   026590
     1            ,(COM3(4533),RANGEM(1)),(COM3(4546),DEFM(1))          CONTR13   026600
     2            ,(COM3(4559),MIDRGM(1)),(COM3(4572),MIDDFM(1))        CONTR14   026610
     3            ,(COM3(4585),TEMPLA(1))                               CONTR15   026620
     4            ,(COM3(4597),TEMPKA(1),FEMPKA(1))                     CONTR16   026630
     5            ,(COM3(4609),RANGEC(1)),(COM3(4622),DEFC(1))          CONTR17   026640
     6            ,(COM3(4635),MIDRGC(1)),(COM3(4648),MIDDFC(1))        CONTR18   026650
     7            ,(COM3(4661),BOUND(1,1))                              CONTR19   026660
     8            ,(COM3(8460),BUFFER(1),ABUFFR(1))                     CONTR20   026670
     9            ,(COM3(8562),HEADER(1)),(COM3(9975),IKRIT)            CONTR21   026680
     A            ,(COM3(9977),SHELX),(COM3(9978),SHELY)                CONTR22   026690
C                                                                       CONTR23   026700
      COMMON /DISC/DISC(15016)                                          CONTR24   026710
      DIMENSION SLIST(5002,2)                                           CONTR25   026720
      EQUIVALENCE (DISC(5005),SLIST(1,1)),(DISC(15011),NSTAPE)          CONTR26   026730
     1           ,(DISC(15013),STAPE)     ,(DISC(15016),SNAME)          CONTR27   026740
C                                                                       CONTR28   026750
      COMMON /CELL/CELL                                                 CONTR29   026760
      COMMON /COLROW/ICOL,IROW,ICOL1,IROW1                              CONTR30   026770
      COMMON /CONSTT/CONSTT(6)                                          CONTR31   026780
      EQUIVALENCE (CONSTT(4),FTTOM),(CONSTT(5),DIMX)                    CONTR32   026790
      COMMON /HOB/IHOB,NHOB,HOB                                         CONTR33   026800
      EQUIVALENCE (IHOB,IPNT)                                           CONTR34   026810
      COMMON /KRIT/KRIT(7)                                              CONTR35   026820
      COMMON /KSYM/KSYM                                                 CONTR36   026830
      COMMON /LEVEL3/LEVEL3(20000)                                      CONTR37   026840
      LEVEL 3, LEVEL3                                                   CONTR38   026850
      COMMON /LIMIT/LIMIT1,LIMIT2,LIMIT3,LIMIT4                         CONTR39   026860
      EQUIVALENCE (LIMIT1,LYMXL),(LIMIT2,LYMXR),(LIMIT3,LYMYB)          CONTR40   026870
     1           ,(LIMIT4,LYMYT)                                        CONTR41   026880
C     NOTE THAT THE 4 LYM-- VARIABLES ARE NOT THE ONES STORED IN COMMON CONTR42   026890
      COMMON /PAGE/LINES,PAGE                                           CONTR43   026900
C                                                                       CONTR44   026910
      DIMENSION ALFA(13),BND(13),BNND(14),IFMT1(2),IFMT2(17)            CONTR45   026920
      EQUIVALENCE (ALFA(13),BLANK),(BND(1),BNND(2),ZERO)                CONTR46   026930
C                                                                       CONTR47   026940
      INTEGER ALFA,BLANK,BOUND,BUFFER,DIMX,HEADER,PAGE,SLIST,SNAME,SPACECONTR48   026950
     1        ,STAPE,TEMPKA,TOP,USED,ZERO                               CONTR49   026960
      REAL MIDDFC,MIDDFM,MIDRGC,MIDRGM                                  CONTR50   026970
C                                                                       CONTR51   026980
      DATA (IFMT2(I),I=1,17)/10H(       131,10HH THE PROB,10HABILITY CO CONTR52D  026990
     1        ,10HNTOURS FOR,10H THIS TARG,10HET, HOB, A,10HND DAMAGE   CONTR53D  027000
     2        ,10HCRITERION ,10HFOLLOW. TH,10HE NUMBERS ,10HARE PD*10.  CONTR54D  027010
     3        ,10H (TRUNCATE,10HD), AND E=,10H.0005, *=1,10H.0)         CONTR55D  027020
     4        ,10H(       131,10H(1H1/  131/                            CONTR56D  027030
      DATA (BNND(I),I=1,14) /-.1, 0., .0005, .0995, .1995, .2995, .3995,CONTR57D  027040
     1 .4995, .5995, .6995, .7995, .8995, .9995, 1.1/                   CONTR58D  027050
      DATA (ALFA(I),I=1,13) /1H0, 1H1, 1H1, 1H2, 1H3, 1H4, 1H5, 1H6, 1H7CONTR59D  027060
     1, 1H8, 1H9, 1H*, 1H /                                             CONTR60D  027070
```

```
      DATA ONE, IONE /1., 1/                                            CONTR61D 027080
C                                                                       CONTR62  027090
      LMXL=LYMXL                                                        CONTR63  027100
      LMXR=LYMXR                                                        CONTR64  027110
      LMYB=LYMYB                                                        CONTR65  027120
      LMYT=LYMYT                                                        CONTR66  027130
      SHLX=SHELX                                                        CONTR67  027140
      SHLY=SHELY                                                        CONTR68  027150
      SPACE=24                                                          CONTR69  027160
      CNSTNT=CELL*FTTOM                                                 CONTR70  027170
      CNSTT2=CNSTNT**2                                                  CONTR71  027180
      J=LMYB                                                            CONTR72  027190
      ASSIGN 250 TO JUMP1                                               CONTR73  027200
      ASSIGN 470 TO JUMP3                                               CONTR74  027210
      IF (KSYM.EQ.2.OR.LMYB.GT.IROW1.OR.LMYT.LE.IROW1) GO TO 10         CONTR75  027220
      ASSIGN 460 TO JUMP3                                               CONTR76  027230
      NEVEL3=(LMYT-LMYB+1)*100                                          CONTR77  027240
      CALL MOVLEV (PK(1,LMYB),LEVEL3(1),NEVEL3)                         CONTR78  027250
   10 J1=J+1                                                            CONTR79  027260
      DO 20 K=1,13                                                      CONTR80  027270
      BOUND(K,1)=99999                                                  CONTR81  027280
      BOUND(K,2)=-99999                                                 CONTR82  027290
      BOUND(K,3)=99999                                                  CONTR83  027300
      BOUND(K,4)=-99999                                                 CONTR84  027310
      TEMPLA(K)=0.                                                      CONTR85  027320
   20 TEMPKA(K)=0                                                       CONTR86  027330
C     GET KILL AREA AND LETHAL AREA FOR EACH PD REGION                  CONTR87  027340
      K=1                                                               CONTR88  027350
      DO 60 KI=LMXL,LMXR                                                CONTR89  027360
      DO 60 KJ=J,LMYT                                                   CONTR90  027370
      KQ=DIMX*(KJ-1)+KI                                                 CONTR91  027380
      IF (PPKK(KQ).LT.BND(K)) GO TO 40                                  CONTR92  027390
   30 IF (PPKK(KQ).LT.BND(K+1)) GO TO 50                                CONTR93  027400
      K=K+1                                                             CONTR94  027410
      GO TO 30                                                          CONTR95  027420
   40 K=K-1                                                             CONTR96  027430
      IF (PPKK(KQ).LT.BND(K)) GO TO 40                                  CONTR97  027440
   50 TEMPLA(K)=TEMPLA(K)+PPKK(KQ)                                      CONTR98  027450
   60 TEMPKA(K)=TEMPKA(K)+1                                             CONTR99  027460
C     GET CONTOURS AND MAXIMUM EXTENT THEREOF FOR TOP AND BOTTOM ROWS   CONT100  027470
   70 I=LMXL                                                            CONT101  027480
      KQ=DIMX*(J-1)+I                                                   CONT102  027490
      K=1                                                               CONT103  027500
   80 IF (PPKK(KQ).GE.BND(K)) GO TO 120                                 CONT104  027510
   90 KUNTR(KQ)=ALFA(K-1)                                               CONT105  027520
      IF (BOUND(K,1).GT.I) BOUND(K,1)=I                                 CONT106  027530
      IF (BOUND(K,2).LT.I) BOUND(K,2)=I                                 CONT107  027540
      IF (BOUND(K,3).GT.J) BOUND(K,3)=J                                 CONT108  027550
      IF (BOUND(K,4).LT.J) BOUND(K,4)=J                                 CONT109  027560
      IF (I.EQ.LMXR) GO TO 130                                          CONT110  027570
      I=I+1                                                             CONT111  027580
      KQ=KQ+1                                                           CONT112  027590
      IF (PPKK(KQ).GE.BND(K)) GO TO 120                                 CONT113  027600
      IF (PPKK(KQ).GE.BND(K-1)) GO TO 90                                CONT114  027610
      DO 100 KK=1,K                                                     CONT115  027620
      KKK=K-KK                                                          CONT116  027630
      IF (PPKK(KQ).GE.BND(KKK-1)) GO TO 110                             CONT117  027640
  100 CONTINUE                                                          CONT118  027650
  110 K=KKK                                                             CONT119  027660
      GO TO 90                                                          CONT120  027670
```

```
  120 K=K+1                                                                CONT121  027680
      IF (K.LE.13) GO TO 80                                                CONT122  027690
  130 IF (J.EQ.LMYT) GO TO 140                                             CONT123  027700
      J=LMYT                                                               CONT124  027710
      GO TO 70                                                             CONT125  027720
C     GET CONTOURS AND MAXIMUM EXTENT THEREOF FOR REST OF ARRAY            CONT126  027730
  140 J2=LMYT-1                                                            CONT127  027740
      IF (J1.GT.J2) GO TO 240                                              CONT128  027750
      DO 230 J=J1,J2                                                       CONT129  027760
      I=LMXL                                                               CONT130  027770
      KQ=DIMX*(J-1)+I                                                      CONT131  027780
      K=1                                                                  CONT132  027790
  150 IF (PPKK(KQ).GE.BND(K)) GO TO 210                                    CONT133  027800
      KUNTR(KQ)=ALFA(K-1)                                                  CONT134  027810
      IF (BOUND(K,1).GT.I) BOUND(K,1)=I                                    CONT135  027820
      IF (BOUND(K,2).LT.I) BOUND(K,2)=I                                    CONT136  027830
      IF (BOUND(K,3).GT.J) BOUND(K,3)=J                                    CONT137  027840
      IF (BOUND(K,4).LT.J) BOUND(K,4)=J                                    CONT138  027850
      IF (I.EQ.LMXR) GO TO 230                                             CONT139  027860
  160 I=I+1                                                                CONT140  027870
      KQ=KQ+1                                                              CONT141  027880
      IF (PPKK(KQ).GE.BND(K)) GO TO 210                                    CONT142  027890
      IF (PPKK(KQ).LT.BND(K-1)) GO TO 170                                  CONT143  027900
      IF (I.EQ.LMXR) GO TO 220                                             CONT144  027910
      KUNTR(KQ)=BLANK                                                      CONT145  027920
      GO TO 160                                                            CONT146  027930
  170 KUNTR(KQ-1)=ALFA(K-1)                                                CONT147  027940
      IF (BOUND(K,1).GT.I-1) BOUND(K,1)=I-1                                CONT148  027950
      IF (BOUND(K,2).LT.I-1) BOUND(K,2)=I-1                                CONT149  027960
      IF (BOUND(K,3).GT.J) BOUND(K,3)=J                                    CONT150  027970
      IF (BOUND(K,4).LT.J) BOUND(K,4)=J                                    CONT151  027980
      II=I                                                                 CONT152  027990
      DO 180 KK=1,K                                                        CONT153  028000
      KKK=K-KK                                                             CONT154  028010
      IF (PPKK(KQ).GE.BND(KKK-1)) GO TO 190                                CONT155  028020
  180 CONTINUE                                                             CONT156  028030
  190 K=KKK                                                                CONT157  028040
  200 IF (I.EQ.LMXR) GO TO 220                                             CONT158  028050
      I=I+1                                                                CONT159  028060
      KQ=KQ+1                                                              CONT160  028070
      IF (PPKK(KQ).LT.BND(K-1)) GO TO 170                                  CONT161  028080
      KUNTR(KQ-1)=BLANK                                                    CONT162  028090
      IF (PPKK(KQ).LT.BND(K)) GO TO 200                                    CONT163  028100
      KONTR(II,J)=ALFA(K-1)                                                CONT164  028110
      IF (BOUND(K,1).GT.II) BOUND(K,1)=II                                  CONT165  028120
      IF (BOUND(K,2).LT.II) BOUND(K,2)=II                                  CONT166  028130
      IF (BOUND(K,3).GT.J) BOUND(K,3)=J                                    CONT167  028140
      IF (BOUND(K,4).LT.J) BOUND(K,4)=J                                    CONT168  028150
      KUNTR(KQ-1)=ALFA(K-1)                                                CONT169  028160
      IF (BOUND(K,1).GT.I-1) BOUND(K,1)=I-1                                CONT170  028170
      IF (BOUND(K,2).LT.I-1) BOUND(K,2)=I-1                                CONT171  028180
      IF (BOUND(K,3).GT.J) BOUND(K,3)=J                                    CONT172  028190
      IF (BOUND(K,4).LT.J) BOUND(K,4)=J                                    CONT173  028200
  210 K=K+1                                                                CONT174  028210
      IF (K.LE.13) GO TO 150                                               CONT175  028220
  220 KUNTR(KQ)=ALFA(K-1)                                                  CONT176  028230
      IF (BOUND(K,1).GT.I) BOUND(K,1)=I                                    CONT177  028240
      IF (BOUND(K,2).LT.I) BOUND(K,2)=I                                    CONT178  028250
      IF (BOUND(K,3).GT.J) BOUND(K,3)=J                                    CONT179  028260
      IF (BOUND(K,4).LT.J) BOUND(K,4)=J                                    CONT180  028270
```

```
  230 CONTINUE
C
  240 ASSIGN 410 TO JUMP2
      GO TO JUMP1, (250,330)
C     OUTPUT PAGE HEADING
  250 LINES=USED(LINES)
      KNTY=LMYT-LMYB+1
      KNTX=LMXR-LMXL+1
      ASSIGN 420 TO JUMP2
      IF (PAGE-LINES.GE.KNTY+14) GO TO 260
      IFMT2(1)=IFMT2(17)
      LINES=0
      GO TO 270
  260 WRITE (6,490)
      LINES=LINES+4
  270 WRITE (6,IFMT2)
      LINES=LINES+1
      IFMT2(1)=IFMT2(16)
C     LABEL ARRAY COLUMNS
      TOP=0
  280 J=0
      DO 290 I=LMXL,LMXR
      J=J+1
      K=I/100
      HEADER(J)=10H
      IF (K.NE.0) ENCODE (10,510,HEADER(J) )K
  290 CONTINUE
      WRITE (6,520) (HEADER(I),I=1,J)
      J=0
      DO 300 I=LMXL,LMXR
      J=J+1
  300 HEADER(J)=(I-(I/100)*100)/10
      WRITE (6,530) (HEADER(I),I=1,J)
      J=0
      DO 310 I=LMXL,LMXR
      J=J+1
  310 HEADER(J)=I-(I/10)*10
      WRITE (6,530) (HEADER(I),I=1,J)
      LINES=LINES+4
C     JUMP OUT IF WE'VE JUST LABELED COLUMNS AT THE BOTTOM OF THE MAP
      IF (TOP.NE.0) GO TO 330
      TOP=1
      ENCODE (20,550,IFMT1(1) )KNTX
      WRITE (6,540)
C     OUTPUT CONTOUR MAP
      DO 320 JJ=1,KNTY
      J=LMYT-JJ+1
      WRITE (6,IFMT1) J,(KONTR(I,J),I=LMXL,LMXR),J
  320 CONTINUE
      LINES=LINES+KNTY+1
C     GO BACK AND LABEL COLUMNS AT BOTTOM OF MAP
      GO TO 280
C     OUTPUT OF MAP IS COMPLETE
  330 N=0
      DO 350 K=1,12
      IF (BOUND(K+1,1).EQ.99999) GO TO 350
      N=N+1
      PKBAR(N)=0.
      TEMPRA(N)=FLOAT(TEMPRA(K))*CNSTT2
      TEMPLA(N)=TEMPLA(K)*CNSTT2
```

```
      IF (FEMPKA(N).NE.0.) PKBAR(N)=TEMPLA(N)/FEMPKA(N)                 CONT241  028880
      DO 340 J=1,4                                                      CONT242  028890
  340 BOUND(N,J)=BOUND(K+1,J)                                           CONT243  028900
      RANGEC(N)=BOUND(N,2)-BOUND(N,1)+1                                 CONT244  028910
      DEFC(N)=BOUND(N,4)-BOUND(N,3)+1                                   CONT245  028920
      RANGEM(N)=RANGEC(N)*CNSTNT                                        CONT246  028930
      DEFM(N)=DEFC(N)*CNSTNT                                            CONT247  028940
      MIDRGC(N)=FLOAT(BOUND(N,1)+BOUND(N,2))/2.-SHLX                    CONT248  028950
      MIDDFC(N)=FLOAT(BOUND(N,3)+BOUND(N,4))/2.-SHLY                    CONT249  028960
      MIDRGM(N)=MIDRGC(N)*CNSTNT                                        CONT250  028970
      MIDDFM(N)=MIDDFC(N)*CNSTNT                                        CONT251  028980
  350 CONTINUE                                                          CONT252  028990
      KK1=1                                                             CONT253  029000
      IF (PKBAR(1).LE..000499999) KK1=2                                 CONT254  029010
      I1=N+1-KK1                                                        CONT255  029020
      SPACE=SPACE+2*I1                                                  CONT256  029030
      CELL1=12.*CELL                                                    CONT257  029040
C     OUTPUT SEQUENCE IS OUTERMOST REGION TO INNERMOST REGION. THIS IS  CONT258  029050
C     USUALLY LOW PK TO HIGH PK ORDER.                                  CONT259  029060
      BUFFER(1)=NSTAPE                                                  CONT260  029070
      BUFFER(2)=IPNT                                                    CONT261  029080
      BUFFER(3)=KSYM                                                    CONT262  029090
      BUFFER(4)=KRIT(IKRIT)                                             CONT263  029100
      BUFFER(5)=LMXL                                                    CONT264  029110
      BUFFER(6)=LMXR                                                    CONT265  029120
      BUFFER(7)=LMYB                                                    CONT266  029130
      BUFFER(8)=LMYT                                                    CONT267  029140
      ABUFFR(9)=SHLX                                                    CONT268  029150
      ABUFFR(10)=SHLY                                                   CONT269  029160
      ABUFFR(11)=HOB                                                    CONT270  029170
      ABUFFR(12)=CELL1                                                  CONT271  029180
      IF (KK1.GT.N) GO TO 370                                           CONT272  029190
      BUFFER(13)=I1                                                     CONT273  029200
      J3=13                                                             CONT274  029210
      DO 360 I=KK1,N                                                    CONT275  029220
      ABUFFR(J3+1)=PKBAR(I)                                             CONT276  029230
      ABUFFR(J3+2)=RANGEM(I)                                            CONT277  029240
      ABUFFR(J3+3)=DEFM(I)                                              CONT278  029250
      ABUFFR(J3+4)=MIDRGM(I)                                            CONT279  029260
      ABUFFR(J3+5)=MIDDFM(I)                                            CONT280  029270
      ABUFFR(J3+6)=TEMPLA(I)                                            CONT281  029280
      ABUFFR(J3+7)=FEMPKA(I)                                            CONT282  029290
  360 J3=J3+7                                                           CONT283  029300
      GO TO 390                                                         CONT284  029310
  370 I1=1                                                              CONT285  029320
      BUFFER(13)=1                                                      CONT286  029330
      ABUFFR(14)=0.                                                     CONT287  029340
      DO 380 J3=15,20                                                   CONT288  029350
  380 ABUFFR(J3)=1.                                                     CONT289  029360
      ABUFFR(19)=0.                                                     CONT290  029370
  390 CALL WRITMS (STAPE,BUFFER(1),J3,NSTAPE,-1,0)                      CONT291  029380
      SLIST(NSTAPE,2)=J3                                                CONT292  029390
      LINES=USED(LINES)                                                 CONT293  029400
      IF (PAGE-LINES.GE.SPACE) GO TO 400                                CONT294  029410
      WRITE (6,480)                                                     CONT295W 029420
      LINES=0                                                           CONT296  029430
      GO TO JUMP2, (410,420)                                            CONT297  029440
  400 WRITE (6,500)                                                     CONT298W 029450
      LINES=LINES+3                                                     CONT299  029460
      GO TO JUMP2, (410,420)                                            CONT300  029470
```

```
  410 WRITE (6,560)                                                       CONT301W 029480
      LINES=LINES+3                                                       CONT302  029490
      ASSIGN 470 TO JUMP3                                                 CONT303  029500
  420 WRITE (6,570) SNAME,NSTAPE                                          CONT304W 029510
      NSTAPE=NSTAPE+1                                                     CONT305  029520
      WRITE (6,560)                                                       CONT306W 029530
      SUMLA=0.                                                            CONT307  029540
      SUMKA=0.                                                            CONT308  029550
      DO 430 KK=KK1,N                                                     CONT309  029560
      K=N-KK+KK1                                                          CONT310  029570
      SUMLA=SUMLA+TEMPLA(K)                                               CONT311  029580
      SUMKA=SUMKA+FEMPKA(K)                                               CONT312  029590
      WRITE (6,590) PKBAR(K),TEMPLA(K),SUMLA,FEMPKA(K),SUMKA              CONT313W 029600
  430 CONTINUE                                                            CONT314  029610
      DO 440 K=KK1,N                                                      CONT315  029620
      RANGEC(K)=RANGEM(K)/CNSTNT                                          CONT316  029630
  440 DEFC(K)=DEFM(K)/CNSTNT                                              CONT317  029640
      WRITE (6,600)                                                       CONT318W 029650
      DO 450 KK=KK1,N                                                     CONT319  029660
      K=N-KK+KK1                                                          CONT320  029670
      WRITE (6,610) PKBAR(K),RANGEM(K),RANGEC(K),MIDRGM(K),MIDREC(K),     CONT321W 029680
     1  DEFM(K),DEFC(K),MIDDFM(K),MIDDFC(K)                               CONT322W 029690
  450 CONTINUE                                                            CONT323  029700
      LINES=LINES+2*(N+KK1)+22                                            CONT324  029710
C                                                                         CONT325  029720
      GO TO JUMP3, (460,470)                                              CONT326  029730
  460 CALL MOVLEV (LEVEL3(1),PK(1,LMY6),NEVEL3)                           CONT327  029740
      ASSIGN 330 TO JUMP1                                                 CONT328  029750
      J=IROW1+1                                                           CONT329  029760
      SPACE=27                                                            CONT330  029770
      GO TO 10                                                            CONT331  029780
C                                                                         CONT332  029790
  470 RETURN                                                              CONT333  029800
C                                                                         CONT334  029810
C                                                                         CONT335  029820
  480 FORMAT (1H1)                                                        CONT336  029830
  490 FORMAT (////)                                                       CONT337  029840
  500 FORMAT (///)                                                        CONT338  029850
  510 FORMAT (I1,9X)                                                      CONT339  029860
  520 FORMAT (1H0,6X,124A1)                                               CONT340  029870
  530 FORMAT (1H ,6X,124I1)                                               CONT341  029880
  540 FORMAT (1H )                                                        CONT342  029890
  550 FORMAT (11H(1H ,I3,3X,,I3,6HA1,I5))                                 CONT343  029900
  560 FORMAT (122H THE FOLLOWING RESULTS CORRESPOND TO THOSE PRECEEDING,CONT344  029910
     1 BUT ARE FOR ROWS OF HIGHER ORDER THAN THE PROJECTILE LOCATION ONLCONT345  029920
     2Y.//)                                                               CONT346  029930
  570 FORMAT (45H THESE DATA ARE ON THE DAMAGE FUNCTION FILE (,A10,18H) CONT347  029940
     1AS RECORD NUMBER,I4///)                                             CONT348  029950
  580 FORMAT (1H ,26X,78HAVERAGE PD WITHIN        LETHAL AREA IN METERS**CONT349  029960
     12       DAMAGE AREA IN METERS**2/1H ,29X,11HEACH REGION/1H ,44X,2(6CONT350  029970
     2X,24HPER REGION    CUMULATIVE)//)                                   CONT351  029980
  590 FORMAT (1H ,F37.3,5X,2(F16.3,F14.3))                                CONT352  029990
  600 FORMAT (///124H        AVERAGE PD WITHIN        RECTANGULARIZED, UNCONT353  030000
     1NORMALIZED RANGE DATA        RECTANGULARIZED, UNNORMALIZED DEFLECTICONT354  030010
     2ON DATA/1H ,10X,11HEACH REGION/1H ,20X,2(11X,38HEXTENT OF REGION  CONT355  030020
     3  LOCATION OF REGION)/1H ,22X,2(28X,21HCENTER WRT PROJECTILE)//1H CONT356  030030
     4,19X,2(7X,2(5X,16HMETERS      CELLS))//)                            CONT357  030040
  610 FORMAT (1H ,F18.3,1X,2(7X,2(F11.3,F10.1)))                          CONT358  030050
      END                                                                 CONT359- 030060
```

```
      SUBROUTINE SHOWW (KK)                                              *2597*17 030070
C     THIS SUBROUTINE OUTPUTS THE LAMBDA ARRAY                           SHOWW 2  030080
C                                                                        SHOWW 3  030090
      COMMON /COM1/COM1(20100)                                           SHOWW 4  030100
      DIMENSION LAMDA(100,200),LAMD(100,200)                             SHOWW 5  030110
      EQUIVALENCE (COM1(1),LAMDA(1,1),LAMD(1,1),DUMMY)                   SHOWW 6  030120
C                                                                        SHOWW 7  030130
      COMMON /CELL/CELL                                                  SHOWW 8  030140
      COMMON /CONSTT/CONSTT(6)                                           SHOWW 9  030150
      EQUIVALENCE (CONSTT(4),FTTOM)                                      SHOWW10  030160
      COMMON /LYMIT/LYMXL,LYMXR,LYMYB,LYMYT                              SHOWW11  030170
      COMMON /PAGE/LINES,PAGE                                            SHOWW12  030180
      COMMON /SHOW/SHOW(25)                                              SHOWW13  030190
C                                                                        SHOWW14  030200
      DIMENSION IFMT1(3),IFMT2(5)                                        SHOWW15  030210
      EQUIVALENCE (AREA,IAREA)                                           SHOWW16  030220
C                                                                        SHOWW17  030230
      INTEGER PAGE,SHOW,USED                                             SHOWW18  030240
      REAL LAMDA                                                         SHOWW19  030250
C                                                                        SHOWW20  030260
      DATA (IFMT1(I),I=1,3)/10H(1H ,I3,2X,10H,20I6,I5) ,10H,20I6,I5) /   SHOWW21D 030270
      DATA (IFMT2(I),I=1,5)/10H(/5H      ,10H    ,20I6 ),10H    ,20I6 )  SHOWW22D 030280
     1       ,10H(1H1/ 5H  ,10H(/5H      /                               SHOWW23D 030290
C                                                                        SHOWW24  030300
      FACTR=10.**SHOW(KK)                                                SHOWW25  030310
      CUTOFF=1./FACTR                                                    SHOWW26  030320
      IAREA=0                                                            SHOWW27  030330
      SUM=0.                                                             SHOWW28  030340
      DO 10 I=LYMXL,LYMXR                                                SHOWW29  030350
      DO 10 J=LYMYB,LYMYT                                                SHOWW30  030360
      IF (LAMDA(I,J).LT.CUTOFF) GO TO 10                                 SHOWW31  030370
      SUM=SUM+LAMDA(I,J)                                                 SHOWW32  030380
      IAREA=IAREA+1                                                      SHOWW33  030390
   10 CONTINUE                                                           SHOWW34  030400
      AREA=FLOAT(IAREA)*(CELL*FTTOM)**2                                  SHOWW35  030410
      IF (AREA.GT..000001) SUM=SUM/AREA                                  SHOWW36  030420
      DO 20 I=LYMXL,LYMXR                                                SHOWW37  030430
      DO 20 J=LYMYB,LYMYT                                                SHOWW38  030440
   20 LAMD(I,J)=LAMDA(I,J)*FACTR                                         SHOWW39  030450
      KNT=LYMYT-LYMYB+1                                                  SHOWW40  030460
      WRITE (6,80) SUM,AREA                                              SHOWW41W 030470
      LINES=LINES+2                                                      SHOWW42  030480
      KNT1=(LYMXR-LYMXL+1)/20                                            SHOWW43  030490
      KEY=LYMXR-LYMXL+1-20*KNT1                                          SHOWW44  030500
      IF (KNT1.EQ.0) GO TO 60                                            SHOWW45  030510
      DO 50 I=1,KNT1                                                     SHOWW46  030520
      LINES=USED(LINES)                                                  SHOWW47  030530
      IF (PAGE-LINES.GE.KNT+5) GO TO 30                                  SHOWW48  030540
      IFMT2(1)=IFMT2(4)                                                  SHOWW49  030550
      LINES=-1                                                           SHOWW50  030560
   30 ISTRT=LYMXL+(I-1)*20                                               SHOWW51  030570
      ISTOP=ISTRT+19                                                     SHOWW52  030580
      WRITE (6,IFMT2) (IICOL,IICOL=ISTRT,ISTOP)                          SHOWW53W 030590
      IFMT2(1)=IFMT2(5)                                                  SHOWW54  030600
      WRITE (6,90)                                                       SHOWW55W 030610
      DO 40 J=1,KNT                                                      SHOWW56  030620
      IIROW=LYMYT+1-J                                                    SHOWW57  030630
      WRITE (6,IFMT1) IIROW,(LAMD(IICOL,IIROW),IICOL=ISTRT,ISTOP),IIROW  SHOWW58W 030640
   40 CONTINUE                                                           SHOWW59  030650
      WRITE (6,IFMT2) (IICOL,IICOL=ISTRT,ISTOP)                          SHOWW60W 030660
```

```
      LINES=LINES+KNT+5                                                 SHOWW61  030670
   50 CONTINUE                                                          SHOWW62  030680
      IF (KEY.EQ.0) RETURN                                              SHOWW63  030690
      LINES=USED(LINES)                                                 SHOWW64  030700
      IF (PAGE-LINES.GE.KNT+5) GO TO 60                                 SHOWW65  030710
      IFMT2(1)=IFMT2(4)                                                 SHOWW66  030720
      LINES=-1                                                          SHOWW67  030730
   60 ISTRT=LYMXL+KNT1*20                                               SHOWW68  030740
      ENCODE (10,100,IFMT1(2) )KEY                                      SHOWW69* 030750
      ENCODE (10,110,IFMT2(2) )KEY                                      SHOWW70* 030760
      WRITE (6,IFMT2) (IICOL,IICOL=ISTRT,LYMXR)                         SHOWW71W 030770
      IFMT2(1)=IFMT2(5)                                                 SHOWW72  030780
      WRITE (6,90)                                                      SHOWW73W 030790
      DO 70 J=1,KNT                                                     SHOWW74  030800
      IIROW=LYMYT+1-J                                                   SHOWW75  030810
      WRITE (6,IFMT1) IIROW,(LAMD(IICOL,IIROW),IICOL=ISTRT,LYMXR),IIROW SHOWW76W 030820
   70 CONTINUE                                                          SHOWW77  030830
      WRITE (6,IFMT2) (IICOL,IICOL=ISTRT,LYMXR)                         SHOWW78W 030840
      LINES=LINES+KNT+5                                                 SHOWW79  030850
      IFMT1(2)=IFMT1(3)                                                 SHOWW80  030860
      IFMT2(2)=IFMT2(3)                                                 SHOWW81  030870
      RETURN                                                            SHOWW82  030880
C                                                                       SHOWW83  030890
   80 FORMAT (14H AN AVERAGE OF,1PE10.3,55H PERFORATIONS PER METER**2 ISSHOWW84  030900
     1 SUSTAINED OVER AN AREA OF,1PE10.3,10H METERS**2/)                SHOWW85  030910
   90 FORMAT (1H )                                                      SHOWW86  030920
  100 FORMAT (1H,,I2,7HI6,I5) )                                         SHOWW87  030930
  110 FORMAT (4H   ,,I2,4HI6 ))                                         SHOWW88  030940
      END                                                               SHOWW89- 030950
```

```
      SUBROUTINE INDEX (ERROR)                                            *3086*18 030960
C                                                                         INDEX 2  030970
      COMMON /COM3/COM3(10000)                                            INDEX 3  030980
      DIMENSION BUFFER(104),ABUFFR(104),CBUFFR(8,13)                      INDEX 4  030990
      EQUIVALENCE (COM3(8460),BUFFER(1),ABUFFR(1))                        INDEX 5  031000
     1           ,(BUFFER(13),NN),(BUFFER(14),CBUFFR(1,1))                INDEX 6  031010
C                                                                         INDEX 7  031020
      COMMON /DISC/DISC(15016)                                            INDEX 8  031030
      DIMENSION PLIST(2502,2),SLIST(5002,2)                               INDEX 9  031040
      EQUIVALENCE (DISC(1),PLIST(1,1)),(DIS**5005),SLIST(1,1))            INDEX10  031050
     1           ,(DISC(15013),STAPE) ,(DISC(15014),TAPE)                 INDEX11  031060
     2           ,(DISC(15015),TNAME) ,(DISC(15016),SNAME)                INDEX12  031070
C                                                                         INDEX13  031080
      COMMON /AAA/AAA(8)                                                  INDEX14  031090
      COMMON /PAGE/LINES,PAGE                                             INDEX15  031100
C                                                                         INDEX16  031110
      DIMENSION IFMT(61)                                                  INDEX17  031120
C                                                                         INDEX18  031130
      INTEGER AAA,BUFFER,ERROR,PAGE,PLIST,SLIST,SNAME,STAPE,TAPE,TNAME    INDEX19  031140
     1       ,USED                                                        INDEX20  031150
C                                                                         INDEX21  031160
      DATA (IFMT(I),I=1,61)/10H(       1H ,10H,6X,12CHRE,10HCORD   POI    INDEX22D 031170
     1       ,10HNT    SYMM,10HETRY     D,10HAMAGE     ,10H LYMXL    L    INDEX23D 031180
     2       ,10HYMXR   LYM,10HYB   LYMYT,10H    PROJEC,10HTILE           INDEX24D 031190
     3       ,10HHOB       ,10H CELL     ,10H  NUMBER/1,10HH ,6X,2(6H     INDEX25D 031200
     4       ,10HNUMBER,3X),10H,21HCHARAC,10HTER   CRIT,10HERION,38X,     INDEX26D 031210
     5       ,10H45HLOCATIO,10HN       (MET,10HERS)    (IN,10HCHES)    OF INDEX27D 031220
     6       ,10H REGIONS//,10H1H ,I10,I9,10H,I10,2I12,,10H316,3H  (,     INDEX28D 031230
     7       ,10HF6.1,1H,,F,10H6.1,1H),F9,10H.2,F10.2,I,10H11,5(/),1H     INDEX29D 031240
     8       ,10H ,53X,23HD,10HAMAGE FUNC,10HTION REGIO,10HNS//1H ,27     INDEX30D 031250
     9       ,10HX,77HPDBAR,10H      LETHA,10HL     DAMAG,10HE     DIMEN  INDEX31D 031260
     A       ,10HSIONS IN M,10HETERS     ,10HOFFSET OF ,10HMIDPOINT W     INDEX32D 031270
     B       ,10HRT/1H ,38X,10H,2(4HAREA,,10HEX),16H   (,10HUNNORMALIZ    INDEX33D 031280
     C       ,10HED),9X,20H,10HPROJECTILE,10H IN METERS,10H/1H ,58X,2     INDEX34D 031290
     D       ,10H(18HRANGE ,10H  DEFLECTI,10HON,8X)//(1,10HH ,F32.3,F     INDEX35D 031300
     E       ,10H11.3,2F10.,10H3,F11.3,F1,10H5.3,F11.3),10H)              INDEX36D 031310
     F       ,10H(       1H ,10H(1H1/  1H /                               INDEX37D 031320
C                                                                         INDEX38  031330
      IF (ERROR.NE.0) GO TO 50                                            INDEX39  031340
      IF (AAA(1).NE.10HDAMAGE FUN.AND.AAA(5).NE.10HDAMAGE FUN) GO TO 40   INDEX40  031350
      WRITE (6,60) SNAME                                                  INDEX41W 031360
      LINES=4                                                             INDEX42  031370
      DO 30 J=1,5000                                                      INDEX43  031380
      N=SLIST(J,2)                                                        INDEX44  031390
      IF (N.EQ.0) GO TO 30                                                INDEX45  031400
      CALL READMS (STAPE,BUFFER(1),N,J)                                   INDEX46  031410
      LINES=USED(LINES)                                                   INDEX47  031420
      IF (PAGE-LINES.GE.20+NN) GO TO 10                                   INDEX48  031430
      IFMT(1)=IFMT(61)                                                    INDEX49  031440
      LINES=0                                                             INDEX50  031450
      GO TO 20                                                            INDEX51  031460
   10 WRITE (6,90)                                                        INDEX52W 031470
      LINES=LINES+6                                                       INDEX53  031480
   20 WRITE (6,IFMT) (BUFFER(I),I=1,8),(ABUFFR(I),I=9,12),BUFFER(13),     INDEX54W 031490
     1  ((CBUFFR(I,K),I=1,7),K=1,NN)                                      INDEX55W 031500
      LINES=LINES+NN+14                                                   INDEX56  031510
      IFMT(1)=IFMT(60)                                                    INDEX57  031520
   30 CONTINUE                                                            INDEX58  031530
   40 IF (AAA(1).NE.10HEXPECTED P.AND.AAA(5).NE.10HEXPECTED P) GO TO 70   INDEX59  031540
   50 WRITE (6,100) TNAME                                                 INDEX60W 031550
```

```
      DO 60 I=1,2500                                                    INDEX61  031560
      IF (PLIST(1,2).EQ.0) GO TO 60                                     INDEX62  031570
      CALL READMS (TAPE,BUFFER(1),19,I)                                 INDEX63  031580
      ABUFFR(4)=ABUFFR(4)*12.                                           INDEX64  031590
      WRITE (6,110) (BUFFER(J),J=1,3),(ABUFFR(J),J=4,5),(BUFFER(J),J=6, INDEX65W 031600
     1 11),(ABUFFR(J),J=12,13),(BUFFER(J),J=14,17),(ABUFFR(J),J=18,19)  INDEX66W 031610
   60 CONTINUE                                                          INDEX67  031620
   70 CALL WRITMS (TAPE,PLIST(1,2),2501,2501,-1,0)                      INDEX68  031630
      CALL WRITMS (STAPE,SLIST(1,2),5001,5001,-1,0)                     INDEX69  031640
      CALL CLOSMS (TAPE)                                                INDEX70  031650
      CALL CLOSMS (STAPE)                                               INDEX71  031660
      WRITE (6,120)                                                     INDEX72W 031670
      STOP                                                              INDEX73  031680
C                                                                       INDEX74  031690
   80 FORMAT (1H1/1H ,36X,42HA TABULATION OF THE DAMAGE FUNCTION FILE (,INDEX75  031700
     1A10,9H) FOLLOWS//)                                                INDEX76  031710
   90 FORMAT (/////)                                                    INDEX77  031720
  100 FORMAT (1H1/1H ,35X,44HAN INDEX OF THE EXPECTED PERFORATIONS FILE INDEX78  031730
     1(,A10,9H) FOLLOWS///1H0,16X,99HRECORD NXX NYY   CELL   HUB IROW ICINDEX79  031740
     2OL LYMYB LYMYT LYMXL LYMXR    A       B  NSIDE NX NY NZ   T  THCK//INDEX80  031750
     3/)                                                                INDEX81  031760
  110 FORMAT (1H ,I21,I5,I4,F7.2,F6.2,3I5,3I6,2F7.1,I4,I5,2I3,F5.1,F5.2)INDEX82  031770
  120 FORMAT (1H1)                                                      INDEX83  031780
      END                                                               INDEX84- 031790
```

```
      SUBROUTINE SNOW                                                   *3170*19 031800
C     THIS AND THE FOLLOWING 4 SUBROUTINES ARE TAKEN FROM SNOW'S        SNOW  2 031810
C     PROGRAM WITH SOME MODIFICATIONS                                   SNOW  3 031820
C                                                                       SNOW  4 031830
      COMMON /COM1/COM1(20100)                                          SNOW  5 031840
      DIMENSION ZZ(25,25,25)                                            SNOW  6 031850
      EQUIVALENCE (COM1(1),ZZ(1,1,1))                                   SNOW  7 031860
C                                                                       SNOW  8 031870
      COMMON /COM3/COM3(10000)                                          SNOW  9 031880
      DIMENSION A(100),B(100),XSSP(25,13),XBB3(25,13),XBB4(25,13)       SNOW 10 031890
     1          ,XDOSET(25,13),XROSET(25,13),BB3(13),BB4(13),DOSET(13)  SNOW 11 031900
     2          ,ROSET(13),NN(25),FNN(25),SSP(13),HT(25),NUMB(25)       SNOW 12 031910
     3          ,FNUMB(25),NUMNN(25),FNUMNN(25),VOSD(25),VOSR(25)       SNOW 13 031920
     4          ,AAAA(10),KEY2(25)                                      SNOW 14 031930
      EQUIVALENCE  (COM3(1),A(1)),(COM3(101),B(1)),(COM3(201),XSSP(1,1))SNOW 15 031940
     1             ,(COM3(526),XBB3(1,1)),(COM3(851),XBB4(1,1))         SNOW 16 031950
     2             ,(COM3(1176),XDOSET(1,1)),(COM3(1501),XROSET(1,1))   SNOW 17 031960
     3             ,(COM3(1826),BB3(1)),(COM3(1839),BB4(1))             SNOW 18 031970
     4             ,(COM3(1852),DOSET(1)),(COM3(1865),ROSET(1))         SNOW 19 031980
     5             ,(COM3(1878),NN(1),FNN(1)),(COM3(1903),SSP(1))       SNOW 20 031990
     6             ,(COM3(1916),HT(1)),(COM3(1941),NUMB(1),FNUMB(1))    SNOW 21 032000
     7             ,(COM3(1966),NUMNN(1),FNUMNN(1)),(COM3(1991),VOSD(1)) SNOW 22 032010
     8             ,(COM3(2016),VOSR(1)),(COM3(2954),AAAA(1))           SNOW 23 032020
     9             ,(COM3(9954),KEY2(1)),(COM3(9981),LU3)               SNOW 24 032030
     A             ,(COM3(9982),LU4),(COM3(9983),N,FN),(COM3(9984),NBS) SNOW 25 032040
     B             ,(COM3(9985),NNN),(COM3(9986),NVEH,FNVEH)            SNOW 26 032050
     C             ,(COM3(9987),NVS,FNVS),(COM3(9988),S1),(COM3(9989),S2)SNOW 27 032060
     D             ,(COM3(9990),SE12),(COM3(9991),SE22),(COM3(9992),SU1) SNOW 28 032070
     E             ,(COM3(9993),SU2),(COM3(9994),SU3),(COM3(9995),SU4)  SNOW 29 032080
     F             ,(COM3(9996),T1),(COM3(9997),T2),(COM3(9998),L1)     SNOW 30 032090
     G             ,(COM3(9999),U2),(COM3(10000),NHTS,FNHTS)            SNOW 31 032100
C                                                                       SNOW 32 032110
      COMMON /DISC/DISC(15016)                                          SNOW 33 032120
      EQUIVALENCE (DISC(15013),STAPE),(DISC(15016),SNAME)               SNOW 34 032130
C                                                                       SNOW 35 032140
      COMMON /AAA/AAA(8)                                                SNOW 36 032150
      COMMON /PAGE/LINES,PAGE                                           SNOW 37 032160
C                                                                       SNOW 38 032170
      DIMENSION IFMT4(36),IFMT5(9),IFMT6(10),IFMT7(9)                   SNOW 39 032180
C                                                                       SNOW 40 032190
      INTEGER AAA,AAAA,PAGE,SNAME,STAPE,USED                            SNOW 41 032200
C                                                                       SNOW 42 032210
      DATA (IFMT4(I),I=1,36)/10H(1H ,6(/),,10H          ,10H            SNOW 43D 032220
     1       ,10H          ,10H          ,10H          ,10H             SNOW 44D 032230
     2       ,10H,10H FOR H,10HOB =,F6.2,,10H4H HAS,I3,,10H18H REGION   SNOW 45D 032240
     3       ,10HS, OF WHIC,10HH,I3,13H A,10HRE UNNESTE,10HD,4(/),1H    SNOW 46D 032250
     4       ,10H,36X,2HPD,,10H12X,17HREG,10HION DIMENS,10HIONS,11X,1   SNOW 47D 032260
     5       ,10H8HOFFSET O,10HF MIDPOINT,10H/1H ,78X,1,10H8HFROM TAR   SNOW 48D 032270
     6       ,10HGET CENTER,10H/1H ,49X,1,10H9HRANGE   ,10H DEFLECTIO   SNOW 49D 032280
     7       ,10HN,10X,18HR,10HANGE    DEF,10HLECTION//(,10H1H ,F39.3,  SNOW 50D 032290
     8       ,10HF15.2,F12.,10H2,F17.2,F1,10H1.2))     ,10H(1H ,6(/),   SNOW 51D 032300
     9       ,10H(1H1/     /                                            SNOW 52D 032310
      DATA (IFMT5(I),I=1,9) /10H(4(/),17H ,10H   VOLLEY ,10HSIZE =,I7,  SNOW 53D 032320
     1       ,10H9I11/1H , ,10H16X,I11,  ,10H9I11/1H , ,10H20X,5I11/)   SNOW 54D 032330
     2       ,10H9I11/1H , ,10H9I11/)     /                             SNOW 55D 032340
      DATA (IFMT6(I),I=1,10)/10H(/6H PD(HG,10HB=,F6.2,3H,10H) =,10F11.  SNOW 56D 032350
     1       ,10H6/1H ,     ,10H20X,10F11.,10H6/1H ,     ,10H24X,5F11.6   SNOW 57D 032360
     2       ,10H)          ,10H6/1H ,    ,10H6)       /                SNOW 58D 032370
      DATA (IFMT7(I),I=1,9) /10H(6(/),1H ,,10H43X,13HTAR,10HGET NUMBER  SNOW 59D 032380
     1       ,10H,I3,13H LO,10HCATED AT (,10H,F7.2,1H,,,10HF7.2,1H))    SNOW 60D 032390
```

```
     2        ,10H(   1H1/  ,10H(6(/),1H ,/                          SNOW 61D 032400
C                                                                    SNOW 62  032410
  10 DECODE (50,430,AAA(2) )FN,FNVEH,FNVS,FNHTS,U1                   SNOW 63* 032420
     N=FN+.1                                                         SNOW 64  032430
     NVEH=FNVEH+.1                                                   SNOW 65  032440
     NVS=FNVS+.1                                                     SNOW 66  032450
     NHTS=FNHTS+.1                                                   SNOW 67  032460
     U2=U1                                                           SNOW 68  032470
     READ (5,510) (AAA(I),I=1,8)                                     SNOW 69R 032480
     IF (AAA(1).EQ.10HSAME CANNO) GO TO 20                           SNOW 70  032490
     DECODE (80,430,AAA(1) )(A(I),I=1,8)                             SNOW 71* 032500
     IF (N.GT.8) READ (5,430) (A(I),I=9,N)                           SNOW 72R 032510
     READ (5,430) (B(I),I=1,N)                                       SNOW 73R 032520
     GO TO 40                                                        SNOW 74  032530
  20 DO 30 I=1,N                                                     SNOW 75  032540
  30 B(I)=-B(I)                                                      SNOW 76  032550
  40 READ (5,510) (AAA(I),I=1,8)                                     SNOW 77R 032560
     IF (AAA(1).EQ.10HSAME TARGE) GO TO 50                           SNOW 78  032570
     DECODE (80,430,AAA(1) )(VOSR(I),I=1,8)                          SNOW 79* 032580
     IF (NVEH.GT.8) READ (5,430) (VOSR(I),I=9,NVEH)                  SNOW 80R 032590
     READ (5,430) (VOSD(I),I=1,NVEH)                                 SNOW 81R 032600
     GO TO 70                                                        SNOW 82  032610
  50 DO 60 I=1,NVEH                                                  SNOW 83  032620
  60 VOSD(I)=-VOSD(I)                                                SNOW 84  032630
  70 READ (5,510) (AAA(I),I=1,8)                                     SNOW 85R 032640
     IF (AAA(1).EQ.10HSAME VOLLE) GO TO 90                           SNOW 86  032650
     DECODE (80,430,AAA(1) )(FNN(I),I=1,8)                           SNOW 87* 032660
     IF (NVS.GT.8) READ (5,430) (FNN(I),I=9,NVS)                     SNOW 88R 032670
     DO 80 I=1,NVS                                                   SNOW 89  032680
  80 NN(I)=FNN(I)+.1                                                 SNOW 90  032690
  90 READ (5,510) (AAA(I),I=1,8)                                     SNOW 91R 032700
     IF (AAA(1).EQ.10HSAME ERROR) GO TO 100                          SNOW 92  032710
     DECODE (50,430,AAA(1) )S1,S2,T1,T2,R                            SNOW 93* 032720
     IF (T1.EQ.0.) T1=.00001                                         SNOW 94  032730
     IF (T2.EQ.0.) T2=.00001                                         SNOW 95  032740
 100 READ (5,510) (AAA(I),I=1,8)                                     SNOW 96R 032750
     WRITE (6,440) N                                                 SNOW 97W 032760
     WRITE (6,450) (A(I),I=1,N)                                      SNOW 98W 032770
     WRITE (6,460)                                                   SNOW 99W 032780
     WRITE (6,450) (B(I),I=1,N)                                      SNOW100W 032790
C    THIS SUBPROGRAM EXPECTS DEFLECTION RIGHT TO BE POSITIVE         SNOW101  032800
     DO 110 I=1,N                                                    SNOW102  032810
 110 B(I)=-B(I)                                                      SNOW103  032820
     IU1=U1+.1                                                       SNOW104  032830
     IU2=U2+.1                                                       SNOW105  032840
     WRITE (6,500) S1,S2,T1,T2,R,IU1,IU2                             SNOW106W 032850
     WRITE (6,490) NVS,(NN(I),I=1,NVS)                               SNOW107W 032860
     WRITE (6,470) NVEH                                              SNOW108W 032870
     WRITE (6,450) (VOSR(I),I=1,NVEH)                                SNOW109W 032880
     WRITE (6,480)                                                   SNOW110W 032890
     WRITE (6,450) (VOSD(I),I=1,NVEH)                                SNOW111W 032900
C    THIS SUBPROGRAM EXPECTS DEFLECTION RIGHT TO BE POSITIVE         SNOW112  032910
     DO 120 I=1,NVEH                                                 SNOW113  032920
 120 VOSD(I)=-VOSD(I)                                                SNOW114  032930
     IF (AAA(1).EQ.10HSAME DAMAG) GO TO 190                          SNOW115  032940
     DO 180 I=1,NHTS                                                 SNOW116  032950
     IF (I.NE.1) READ (5,510) (AAA(J),J=1,8)                         SNOW117R 032960
     KEY2(I)=10HCARDS                                                SNOW118  032970
     DECODE (10,520,AAA(1) )(AAAA(J),J=1,10)                         SNOW119* 032980
     ASSIGN 130 TO JUMP                                              SNOW120  032990
```

```
      DO 140 J=1,10                                                     SNOW121  033000
      IF (AAAA(J).GT.1HZ) GO TO 140                                     SNOW122  033010
      GO TO JUMP, (130,150)                                             SNOW123  033020
  130 IF (AAAA(J).NE.1HE) GO TO 150                                     SNOW124  033030
      ASSIGN 150 TO JUMP                                                SNOW125  033040
  140 CONTINUE                                                          SNOW126  033050
      GO TO 160                                                         SNOW127  033060
  150 CALL INPUT (I)                                                    SNOW128  033070
      GO TO 180                                                         SNOW129  033080
  160 DECODE (30,430,AAA(1) )HT(I),FNUMB(I),FNUMNN(I)                   SNOW130* 033090
      NUMB(I)=FNUMB(I)+.1                                               SNOW131  033100
      NUMNN(I)=FNUMNN(I)+.1                                             SNOW132  033110
      NBS=NUMB(I)                                                       SNOW133  033120
      DO 170 J=1,NBS                                                    SNOW134  033130
      READ (5,430) XSSP(I,J),XBB3(I,J),XBB4(I,J),XROSET(I,J),XDOSET(I,J)SNOW135R 033140
  170 CONTINUE                                                          SNOW136  033150
  180 CONTINUE                                                          SNOW137  033160
  190 LINES=41+4*((N-1)/10+(NVEH-1)/10)                                 SNOW138  033170
      DO 230 I=1,NHTS                                                   SNOW139  033180
      LAP1=NUMB(I)                                                      SNOW140  033190
      LINES=USED(LINES)                                                 SNOW141  033200
      IF (PAGE-LINES.GE.LAP1+14) GO TO 200                              SNOW142  033210
      IFMT4(1)=IFMT4(36)                                                SNOW143  033220
      LINES=-6                                                          SNOW144  033230
  200 IF (KEY2(I).NE.10HCARDS     ) GO TO 210                           SNOW145  033240
      ENCODE (60,530,IFMT4(2) )                                         SNOW146* 033250
      GO TO 220                                                         SNOW147  033260
  210 ENCODE (60,540,IFMT4(2) )KEY2(I),SNAME                            SNOW148* 033270
  220 WRITE (6,IFMT4) HT(I),NUMB(I),NUMNN(I),(XSSP(I,J),XBB3(I,J),XBB4(ISNOW149W 033280
     1 ,J),XROSET(I,J),XDOSET(I,J),J=1,LAP1)                            SNOW150W 033290
      LINES=LINES+14+LAP1                                               SNOW151  033300
      IFMT4(1)=IFMT4(35)                                                SNOW152  033310
      DO 230 J=1,LAP1                                                   SNOW153  033320
      XROSET(I,J)=-XROSET(I,J)                                          SNOW154  033330
      XDOSET(I,J)=-XDOSET(I,J)                                          SNOW155  033340
  230 XSSP(I,J)=XSSP(I,J)*R                                             SNOW156  033350
      IF (NVS.GT.20) GO TO 240                                          SNOW157  033360
      IFMT5(6)=IFMT5(9)                                                 SNOW158  033370
      IFMT6(6)=IFMT6(10)                                                SNOW159  033380
      IF (NVS.GT.10) GO TO 240                                          SNOW160  033390
      IFMT5(4)=IFMT5(9)                                                 SNOW161  033400
      IFMT6(4)=IFMT6(10)                                                SNOW162  033410
  240 LINEINC=13+2*NHTS+(NHTS+1)*((NVS-1)/10)                           SNOW163  033420
      DO 390 K=1,NVEH                                                   SNOW164  033430
      SU1=VDSR(K)                                                       SNOW165  033440
      SU2=VDSD(K)                                                       SNOW166  033450
      TU1=T1*U1                                                         SNOW167  033460
      TU2=T2*U2                                                         SNOW168  033470
      SSU3=0.                                                           SNOW169  033480
      SSU4=0.                                                           SNOW170  033490
      AJMX=ABS(SU1-A(1))                                                SNOW171  033500
      BJMX=ABS(SU2-B(1))                                                SNOW172  033510
      IF (N.EQ.1) GO TO 260                                             SNOW173  033520
      DO 250 J=2,N                                                      SNOW174  033530
      ASU1=ABS(SU1-A(J))                                                SNOW175  033540
      BSU2=ABS(SU2-B(J))                                                SNOW176  033550
      IF (ASU1.GT.AJMX) AJMX=ASU1                                       SNOW177  033560
      IF (BSU2.GT.BJMX) BJMX=BSU2                                       SNOW178  033570
  250 CONTINUE                                                          SNOW179  033580
  260 DO 270 J=1,N                                                      SNOW180  033590
```

```
      NJ1=N-J+1                                          SNOW181  033600
      SSU3=SSU3+ABS(A(NJ1)+A(J))                         SNOW182  033610
      SSU4=SSU4+ABS(B(NJ1)+B(J))                         SNOW183  033620
  270 CONTINUE                                           SNOW184  033630
      SU3=SU1+SSU3                                       SNOW185  033640
      SU4=SU2+SSU4                                       SNOW186  033650
      U3=U1                                              SNOW187  033660
      IF (SU3.NE.0.) U3=2.*U3                            SNOW188  033670
      U4=U2                                              SNOW189  033680
      IF (SU4.NE.0.) U4=2.*U4                            SNOW190  033690
      LU3=U3                                             SNOW191  033700
      LU4=U4                                             SNOW192  033710
      SE12=2./U1                                         SNOW193  033720
      AK=U3                                              SNOW194  033730
      GO TO 290                                          SNOW195  033740
  280 AK=AK-1.                                           SNOW196  033750
  290 AX=0.                                              SNOW197  033760
      IF (AK.EQ.0.) GO TO 300                            SNOW198  033770
      AX=2.*AK-1.                                        SNOW199  033780
      IF (SU3.NE.0.) AX=AX-2.*U1                         SNOW200  033790
      AX=AX*SE12                                         SNOW201  033800
  300 AA=.5-ERF(AX)                                      SNOW202  033810
      IF (AA.GT..0001) GO TO 310                         SNOW203  033820
      GO TO 280                                          SNOW204  033830
  310 SE3=(SE12+AX)/U1                                   SNOW205  033840
      SE22=2./U2                                         SNOW206  033850
      AK=U4                                              SNOW207  033860
      GO TO 330                                          SNOW208  033870
  320 AK=AK-1.                                           SNOW209  033880
  330 AY=0.                                              SNOW210  033890
      IF (AK.EQ.0.) GO TO 340                            SNOW211  033900
      AY=2.*AK-1.                                        SNOW212  033910
      IF (SU4.NE.0.) AY=AY-2.*U2                         SNOW213  033920
      AY=AY*SE22                                         SNOW214  033930
  340 AA=.5-ERF(AY)                                      SNOW215  033940
      IF (AA.GT..0001) GO TO 350                         SNOW216  033950
      GO TO 320                                          SNOW217  033960
  350 LINES=USED(LINES)                                  SNOW218  033970
      IF (PAGE-LINES.GE.LINEINC) GO TO 360               SNOW219  033980
      IFMT7(1)=IFMT7(8)                                  SNOW220  033990
      LINES=0                                            SNOW221  034000
  360 VOSD(K)=-VOSD(K)                                   SNOW222  034010
      WRITE (6,IFMT7) K,VOSR(K),VOSD(K)                  SNOW223W 034020
      VOSD(K)=-VOSD(K)                                   SNOW224  034030
      IFMT7(1)=IFMT7(9)                                  SNOW225  034040
      WRITE (6,IFMT5) (NN(I),I=1,NVS)                    SNOW226W 034050
      DO 380 L=1,NHTS                                    SNOW227  034060
      NBS=NUMB(L)                                        SNOW228  034070
      NNN=NUMNN(L)                                       SNOW229  034080
      DO 370 JJ=1,NBS                                    SNOW230  034090
      BB3(JJ)=XBB3(L,JJ)                                 SNOW231  034100
      BB4(JJ)=XBB4(L,JJ)                                 SNOW232  034110
      SSP(JJ)=XSSP(L,JJ)                                 SNOW233  034120
      ROSET(JJ)=XROSET(L,JJ)                             SNOW234  034130
  370 DOSET(JJ)=XDOSET(L,JJ)                             SNOW235  034140
      SE4=(SE22+AY)/U2                                   SNOW236  034150
      GSE1=(AJMX+.5*BB3(NBS)+4.*S1)/TU1                  SNOW237  034160
      GSE2=(BJMX+.5*BB4(NBS)+4.*S2)/TU2                  SNOW238  034170
      SE12=AMIN1(SE3,GSE1)*.5                            SNOW239  034180
      SE22=AMIN1(SE4,GSE2)*.5                            SNOW240  034190
```

```
      CALL A50 (K,L)                                                    SNOW241  034200
      WRITE (6,IFMT6) HT(L),(ZZ(K,I,L),I=1,NVS)                         SNOW242W 034210
  380 CONTINUE                                                          SNOW243  034220
      LINES=LINES+LINEINC                                               SNOW244  034230
  390 CONTINUE                                                          SNOW245  034240
  400 READ (5,510) (AAA(I),I=1,8)                                       SNOW246R 034250
      IF (AAA(1).NE.10HSAME HOB O.AND.AAA(1).NE.10HHOB DISTRI.AND.AAA(1)SNOW247  034260
     1.NE.10HOPTIMIZE  .AND.AAA(1).NE.10HALL HOBS  ) GO TO 410          SNOW248  034270
      CALL CMBN                                                         SNOW249  034280
      GO TO 400                                                         SNOW250  034290
  410 IFMT5(4)=IFMT5(8)                                                 SNOW251  034300
      IFMT5(6)=IFMT5(8)                                                 SNOW252  034310
      IFMT6(4)=IFMT6(9)                                                 SNOW253  034320
      IFMT6(6)=IFMT6(9)                                                 SNOW254  034330
      IF (AAA(1).NE.10HENCOUNTER .AND.AAA(1).NE.10H ENCOUNTER) RETURN   SNOW255  034340
      DO 420 I=1,NHTS                                                   SNOW256  034350
      LAP1=NUMB(I)                                                      SNOW257  034360
      DO 420 J=1,LAP1                                                   SNOW258  034370
      XROSET(I,J)=-XROSET(I,J)                                          SNOW259  034380
      XDOSET(I,J)=-XDOSET(I,J)                                          SNOW260  034390
  420 XSSP(I,J)=XSSP(I,J)/R                                             SNOW261  034400
      GO TO 10                                                          SNOW262  034410
C                                                                       SNOW263  034420
  430 FORMAT (8F10.3)                                                   SNOW264  034430
  440 FORMAT (1H1/10H THERE ARE,I3,7H CANNON//62HO      THE AIMING POINT SNOW265  034440
     1RANGE COORDINATES FOR THE CANNON FOLLOW)                          SNOW266  034450
  450 FORMAT (1HO,7X,10F10.2)                                           SNOW267  034460
  460 FORMAT (/67HO      THE AIMING POINT DEFLECTION COORDINATES FOR THE SNOW268  034470
     1CANNON FOLLOW)                                                    SNOW269  034480
  470 FORMAT (//10HOTHERE ARE,I3,8H TARGETS//65HO      THE RANGE COORDINASNOW270  034490
     1TES FOR THE CENTERS OF THE TARGETS FOLLOW)                        SNOW271  034500
  480 FORMAT (/70HO      THE DEFLECTION COORDINATES FOR THE CENTERS OF THSNOW272  034510
     1E TARGETS FOLLOW)                                                 SNOW273  034520
  490 FORMAT (//10HOTHERE ARE,I3,85H DIFFERENT VOLLEY SIZES. THE NUMBERSSNOW274  034530
     1 OF ROUNDS PER CANNON FOR THE VOLLEY SIZES FOLLOW/1HO,25I5)       SNOW275  034540
  500 FORMAT (//54HOTHE ROUND-TO-ROUND SIGMAS IN RANGE AND DEFLECTION ARSNOW276  034550
     1E,F7.2,4H AND,F7.2,13H RESPECTIVELY/52HOTHE AIMING ERROR SIGMAS INSNOW277  034560
     2 RANGE AND DEFLECTION ARE,F7.2,4H AND,F7.2,13H RESPECTIVELY/24HOTHSNOW278  034570
     3E FUZE RELIABILITY IS,F6.3/10HOTHERE ARE,I3,1H*,I2,51H INTEGRATIONSNOW279  034580
     4 STEPS IN EACH QUADRANT FOR EACH TARGET)                          SNOW280  034590
  510 FORMAT (8A10)                                                     SNOW281  034600
  520 FORMAT (10A1)                                                     SNOW282  034610
  530 FORMAT (60H1H ,22X,30HTHE DAMAGE FUNCTION FROM CARDS              SNOW283  034620
     1     )                                                            SNOW284  034630
  540 FORMAT (33H1H ,14X,47HDAMAGE FUNCTION NUMBER,I4,11H FROM FILE ,A10SNOW285  034640
     1,2H  )                                                            SNOW286  034650
      END                                                               SNOW287- 034660
```

```
      SUBROUTINE A50 (KKDUM,LLDUM)
C
      COMMON /COM1/COM1(20100)
      DIMENSION ZZ(25,25,25)
      EQUIVALENCE (COM1(1),ZZ(1,1,1))
C
      COMMON /COM3/COM3(10000)
      DIMENSION NN(25),SSSUM(25),ZZSUM(25)
      EQUIVALENCE  (COM3(1878),NN(1)),(COM3(2041),SSSUM(1))
     1            ,(COM3(2066),ZZSUM(1)),(COM3(9981),LU3)
     2            ,(COM3(9982),LU4),(COM3(9987),NVS),(COM3(9990),SE12)
     3            ,(COM3(9991),SE22),(COM3(9994),SU3),(COM3(9995),SU4)
     4            ,(COM3(9998),U1),(COM3(9999),U2)
C
      KK=KKDUM
      LL=LLDUM
      DO 10 I=1,NVS
   10 ZZSUM(I)=0.0
      DO 60 J=1,LU3
      DO 20 I=1,NVS
   20 SSSUM(I)=0.0
      X=2.*FLOAT(J)-1.
      IF (SU3.NE.0.) X=X-2.*U1
      X=X*SE12
      DO 40 K=1,LU4
      Y=2.*FLOAT(K)-1.
      IF (SU4.NE.0.) Y=Y-2.*U2
      Y=Y*SE22
      PPK6=PK6(X,Y)
      AF=ERF(Y+SE22)-ERF(Y-SE22)
      DO 30 I=1,NVS
   30 SSSUM(I)=SSSUM(I)+(1.0-PPK6**NN(I))*AF
   40 CONTINUE
      AF=ERF(X+SE12)-ERF(X-SE12)
      DO 50 I=1,NVS
   50 ZZSUM(I)=ZZSUM(I)+SSSUM(I)*AF
   60 CONTINUE
      DO 70 I=1,NVS
   70 ZZ(KK,I,LL)=ZZSUM(I)
      IF (SU3.NE.0.) GO TO 90
      DO 80 I=1,NVS
   80 ZZ(KK,I,LL)=2.*ZZ(KK,I,LL)
   90 IF (SU4.NE.0.) GO TO 110
      DO 100 I=1,NVS
  100 ZZ(KK,I,LL)=2.*ZZ(KK,I,LL)
  110 RETURN
      END
```

```
      FUNCTION PK6 (XXDUM,YYDUM)
C
      COMMON /COM3/COM3(10000)
      DIMENSION A(100),B(100),BB3(13),BB4(13),DOSET(13),ROSET(13)
     1          ,SSP(13),XUA(100),YUB(100)
      EQUIVALENCE  (COM3(1),A(1)),(COM3(101),B(1)),(COM3(1826),BB3(1))
     1            ,(COM3(1839),BB4(1)),(COM3(1852),DOSET(1))
     2            ,(COM3(1865),ROSET(1)),(COM3(1903),SSP(1))
     3            ,(COM3(2091),XUA(1)),(COM3(2191),YUB(1))
     4            ,(COM3(9979),B1),(COM3(9980),B2),(COM3(9983),N)
     5            ,(COM3(9984),NBS),(COM3(9985),NNN),(COM3(9988),S1)
     6            ,(COM3(9989),S2),(COM3(9992),SU1),(COM3(9993),SU2)
     7            ,(COM3(9996),T1),(COM3(9997),T2)
C
      XX=XXDUM
      YY=YYDUM
      PK6=1.
      DO 10 J=1,N
      XUA(J)=XX*T1+SU1-A(J)
      YUB(J)=YY*T2+SU2-B(J)
   10 CONTINUE
      DO 20 J=1,N
      POLD=0.0
      DO 20 I=1,NBS
      XTA=XUA(J)-ROSET(I)
      YTA=YUB(J)-DOSET(I)
      B1=.5*BB3(I)
      B2=.5*BB4(I)
      PX1=SMF(XTA,B1,S1)
      PY1=SMF(YTA,B2,S2)
      IF (I.LE.NNN) GO TO 30
      PK6=PK6*(1.-(PX1*PY1-POLD)*SSP(I))
      POLD=PX1*PY1
   20 CONTINUE
      RETURN
   30 PK6=PK6*(1.-PX1*PY1*SSP(I))
      POLD=POLD+PX1*PY1
      GO TO 20
      END
```

```
      FUNCTION SMF (OX,QY,S)                                                *3543*22 035530
      IF (S.EQ.0.) GO TO 10                                                 SMF   2  035540
      SMF=ERF((OX+CY)/S)-ERF((OX-CY)/S)                                     SMF   3  035550
      GO TO 20                                                              SMF   4  035560
   10 SMF=0.                                                                SMF   5  035570
      IF (ABS(OX).LE.QY) SMF=1.                                             SMF   6  035580
   20 RETURN                                                                SMF   7  035590
      END                                                                   SMF   8- 035600
```

```
      FUNCTION ERF (X)                                                      *3551*23 035610
C                                                                           ERF   2  035620
      DATA ROOT2 /.70710678I2/                                              ERF   3D C35630
C     ROOT2=SQRT(1./2.)                                                     ERF   4  035640
C                                                                           ERF   5  035650
      A=ABS(X)*ROOT2                                                        ERF   6  035660
      A2=A*A                                                                ERF   7  035670
      IF (A.GT.4.17) GO TO 10                                               ERF   8  035680
      GO TO 20                                                              ERF   9  035690
   10 ERF=.5                                                                ERF  10  035700
      GO TO 50                                                              ERF  11  035710
   20 IF (A.GT.1.51) GO TO 30                                               ERF  12  035720
      GO TO 40                                                              ERF  13  035730
   30 ERF=.5-EXP(-A2)*.282094792*(A/(A2+.5-(.5/(A2+2.5-(3./(A2+4.5-(7.5 ERF  14  035740
     1 /(A2+6.5-(10.803/(A2+4.269)))))))))                                  ERF  15  035750
      GO TO 50                                                              ERF  16  035760
   40 ERF=A*.564189584*(1.-1.0261*A2/(A2+10.216-(167.17/(A2+9.8103          ERF  17  035770
     1 +(201.39/(A2+11.57+(31.228/(A2-1.773+(64.229759/(A2+5.5777245)))ERF  18  035780
     2 )))))))                                                              ERF  19  035790
   50 IF (X.LT.0.) ERF=-ERF                                                 ERF  20  035800
      RETURN                                                                ERF  21  035810
      END                                                                   ERF  22- 035820
```

```
      SUBROUTINE INPUT (IIIII)
C
C     THIS SUBROUTINE GENERATES A COMPLETELY NESTED DAMAGE FUNCTION FOR
C     THIS PROGRAM FROM THE DAMAGE FUNCTION DATA STORED ON THE DAMAGE
C     FUNCTION FILE BY SUBROUTINE CONTR
C
      COMMON /COM3/COM3(10000)
      DIMENSION QQQ1(25,13),XSSP(325),PKBAR(325),QQQ2(25,13),MAXRG(325)
     1          ,RGM(325),XBB3(325),QQQ3(25,13),MAXDF(325),DFM(325)
     2          ,XBB4(325),QQQ4(25,13),MINDF(325),MIDDFM(325)
     3          ,XDOSET(325),QQQ5(25,13),MINRG(325),MIDRGM(325)
     4          ,XROSET(325),HT(25),NUMB(25),NUMNN(25),QLA(25,13)
     5          ,LA(325),QKA(25,13),KA(325),STEPS2(13),FSTEPS(13)
     6          ,BUFFER(13),ABUFFR(13),BUFFR2(7,13),ABFFR2(7,13)
     7          ,KEY2(25)
      EQUIVALENCE  (COM3(201),QQQ1(1,1),XSSP(1),PKBAR(1))
     1             ,(COM3(526),QQQ2(1,1),MAXRG(1),RGM(1),XBB3(1))
     2             ,(COM3(851),QQQ3(1,1),MAXDF(1),DFM(1),XBB4(1))
     3             ,(COM3(1176),QQQ4(1,1),MINDF(1),MIDDFM(1),XDOSET(1))
     4             ,(COM3(1501),QQQ5(1,1),MINRG(1),MIDRGM(1),XROSET(1))
     5             ,(COM3(1916),HT(1)),(COM3(1941),NUMB(1))
     6             ,(COM3(1966),NUMNN(1)),(COM3(2291),QLA(1,1),LA(1))
     7             ,(COM3(2616),QKA(1,1),KA(1))
     8             ,(COM3(2941),STEPS2(1),FSTEPS(1))
     9             ,(COM3(3446),BUFFER(1),ABUFFR(1),KK),(BUFFER(13),JJ)
     A             ,(COM3(3459),BUFFR2(1,1),ABFFR2(1,1))
     B             ,(COM3(9954),KEY2(1)),(COM3(9984),NBS,NSTEPS)
C
      COMMON /DISC/DISC(15016)
      DIMENSION SLIST(5002,2)
      EQUIVALENCE (DISC(5005),SLIST(1,1)),(DISC(15013),STAPE)
C
      COMMON /AAA/AAA(8)
      COMMON /PAGE/LINES,PAGE
C
      EQUIVALENCE (FILE2,FFILE2)
C
      INTEGER AAA,BUFFER,BUFFR2,FILE2,SLIST
     1         ,STAPE,STEPS1,STEPS2,SUB1,SUB2
      REAL KA,LA,LIM1,LIM2,LIM3,LIM4,LIM5,LIM6,LIM7,LIM8,MAXDF,MAXRG
     1     ,MIDDFM,MIDRGM,MINDF,MINRG,NORM,NORM1,NORM2
C
      I=IIIII
      DECODE (60,220,AAA(3) )FFILE2,(FSTEPS(J),J=1,11)
      FILE2=FFILE2+.1
      DO 10 J=1,11
   10 STEPS2(J)=FSTEPS(J)+.1
      KEY2(I)=FILE2
      CALL READMS (STAPE,BUFFER(1),SLIST(FILE2,2),FILE2)
      HT(I)=ABUFFR(11)
      NSTEPS=BUFFER(13)
      DO 20 J=1,NSTEPS
      QQQ1(I,J)=ABFFR2(1,J)
      QQQ2(I,J)=ABFFR2(2,J)
      QQQ3(I,J)=ABFFR2(3,J)
      QQQ5(I,J)=ABFFR2(4,J)
      QQQ4(I,J)=ABFFR2(5,J)
      QLA(I,J)=ABFFR2(6,J)
   20 QKA(I,J)=ABFFR2(7,J)
      IF (NSTEPS.EQ.1) GO TO 40
```

```
C     CONTR OUTPUTS REGIONS FROM OUTERMOST TO INNERMOST. HERE THE ORDER INPUT61   036430
C     IS REVERSED.                                                      INPUT62   036440
      K=NSTEPS/2                                                        INPUT63   036450
      DO 30 J=1,K                                                       INPUT64   036460
      SUB1=(J-1)*25+I                                                   INPUT65   036470
      SUB2=(NSTEPS-J)*25+I                                              INPUT66   036480
      TEMP=PKBAR(SUB2)                                                  INPUT67   036490
      PKBAR(SUB2)=PKBAR(SUB1)                                           INPUT68   036500
      PKBAR(SUB1)=TEMP                                                  INPUT69   036510
      TEMP=RGM(SUB2)                                                    INPUT70   036520
      RGM(SUB2)=RGM(SUB1)                                               INPUT71   036530
      RGM(SUB1)=TEMP                                                    INPUT72   036540
      TEMP=DFM(SUB2)                                                    INPUT73   036550
      DFM(SUB2)=DFM(SUB1)                                               INPUT74   036560
      DFM(SUB1)=TEMP                                                    INPUT75   036570
      TEMP=MIDRGM(SUB2)                                                 INPUT76   036580
      MIDRGM(SUB2)=MIDRGM(SUB1)                                         INPUT77   036590
      MIDRGM(SUB1)=TEMP                                                 INPUT78   036600
      TEMP=MIDDFM(SUB2)                                                 INPUT79   036610
      MIDDFM(SUB2)=MIDDFM(SUB1)                                         INPUT80   036620
      MIDDFM(SUB1)=TEMP                                                 INPUT81   036630
      TEMP=LA(SUB2)                                                     INPUT82   036640
      LA(SUB2)=LA(SUB1)                                                 INPUT83   036650
      LA(SUB1)=TEMP                                                     INPUT84   036660
      TEMP=KA(SUB2)                                                     INPUT85   036670
      KA(SUB2)=KA(SUB1)                                                 INPUT86   036680
   30 KA(SUB1)=TEMP                                                     INPUT87   036690
   40 DO 50 J=1,NSTEPS                                                  INPUT88   036700
      SUB1=(J-1)*25+I                                                   INPUT89   036710
      MINRG(SUB1)=MIDRGM(SUB1)-RGM(SUB1)/2.                             INPUT90   036720
      MAXRG(SUB1)=MINRG(SUB1)+RGM(SUB1)                                 INPUT91   036730
      MINDF(SUB1)=MIDDFM(SUB1)-DFM(SUB1)/2.                             INPUT92   036740
   50 MAXDF(SUB1)=MINDF(SUB1)+DFM(SUB1)                                 INPUT93   036750
      STEPS1=0                                                          INPUT94   036760
      IF (STEPS2(1).NE.0) GO TO 70                                      INPUT95   036770
      DO 60 J=1,NSTEPS                                                  INPUT96   036780
   60 STEPS2(J)=1                                                       INPUT97   036790
   70 DO 80 J=1,NSTEPS                                                  INPUT98   036800
      IF (STEPS2(J).EQ.0) GO TO 90                                      INPUT99   036810
   80 STEPS1=STEPS1+STEPS2(J)                                           INPU100   036820
C     STEPS1 IS THE NUMBER OF INPUT REGIONS USED                        INPU101   036830
      J=NSTEPS+1                                                        INPU102   036840
   90 J=J-1                                                             INPU103   036850
C     J IS THE NUMBER OF OUTPUT REGIONS                                 INPU104   036860
      IF (STEPS1.EQ.NSTEPS) GO TO 100                                   INPU105   036870
      WRITE (6,230) NSTEPS,HT(I),STEPS1                                 INPU106W  036880
      LINES=LINES+2                                                     INPU107   036890
      IF (STEPS1.LT.NSTEPS) GO TO 100                                   INPU108   036900
      WRITE (6,240)                                                     INPU109W  036910
      CALL INDEX (1)                                                    INPU110   036920
  100 NUMB(I)=J                                                         INPU111   036930
C     NUMB(I) IS THE NUMBER OF OUTPUT REGIONS FOR HT(I)                 INPU112   036940
      NUMNN(I)=0                                                        INPU113   036950
      L2=0                                                              INPU114   036960
      DO 120 K=1,J                                                      INPU115   036970
C     LOOP ON OUTPUT REGIONS                                            INPU116   036980
      L1=L2+1                                                           INPU117   036990
      L2=L2+STEPS2(K)                                                   INPU118   037000
      SUB1=(K-1)*25+I                                                   INPU119   037010
      SUB2=(L1-1)*25+I                                                  INPU120   037020
```

```
      MINRG(SUB1)=MINRG(SUB2)
      MAXRG(SUB1)=MAXRG(SUB2)
      MINDF(SUB1)=MINDF(SUB2)
      MAXDF(SUB1)=MAXDF(SUB2)
      LA(SUB1)=LA(SUB2)
      KA(SUB1)=KA(SUB2)
      IF (L1.EQ.L2) GO TO 120
      L11=L1+1
      DO 110 L=L11,L2
C     LOOP ON INPUT REGIONS BEING COMBINED
      SUB2=(L-1)*25+I
      IF (MINRG(SUB2).LT.MINRG(SUB1)) MINRG(SUB1)=MINRG(SUB2)
      IF (MAXRG(SUB2).GT.MAXRG(SUB1)) MAXRG(SUB1)=MAXRG(SUB2)
      IF (MINDF(SUB2).LT.MINDF(SUB1)) MINDF(SUB1)=MINDF(SUB2)
      IF (MAXDF(SUB2).GT.MAXDF(SUB1)) MAXDF(SUB1)=MAXDF(SUB2)
      LA(SUB1)=LA(SUB1)+LA(SUB2)
  110 KA(SUB1)=KA(SUB1)+KA(SUB2)
  120 XSSP(SUB1)=LA(SUB1)/KA(SUB1)
      NBS=J
      NORM1=0.
      DO 130 J=1,NBS
      SUB1=(J-1)*25+I
      XBB3(SUB1)=MAXRG(SUB1)-MINRG(SUB1)
      XBB4(SUB1)=MAXDF(SUB1)-MINDF(SUB1)
      XROSET(SUB1)=MINRG(SUB1)+XBB3(SUB1)/2.
      XDOSET(SUB1)=MINDF(SUB1)+XBB4(SUB1)/2.
      NORM1=NORM1+KA(SUB1)
      NORM2=XBB3(SUB1)*XBB4(SUB1)
      NORM=SQRT(NORM1/NORM2)
      XBB3(SUB1)=XBB3(SUB1)*NORM
  130 XBB4(SUB1)=XBB4(SUB1)*NORM
      DO 190 J=2,NBS
      SUB1=(J-1)*25+I
      SUB2=(J-2)*25+I
      NORM2=XBB3(SUB1)*XBB4(SUB1)
      IF (XBB4(SUB1).GT.XBB4(SUB2)) GO TO 140
      XBB4(SUB1)=XBB4(SUB2)+.0000001
      XBB3(SUB1)=NORM2/XBB4(SUB1)
      GO TO 150
  140 IF (XBB3(SUB1).GT.XBB3(SUB2)) GO TO 150
      XBB3(SUB1)=XBB3(SUB2)+.0000001
      XBB4(SUB1)=NORM2/XBB3(SUB1)
  150 LIM1=XDOSET(SUB2)-XBB4(SUB2)/2.
      LIM5=XDOSET(SUB1)-XBB4(SUB1)/2.
      IF (LIM1.GT.LIM5) GO TO 160
      XDOSET(SUB1)=XDOSET(SUB1)-LIM5+LIM1-.00000001
      GO TO 170
  160 LIM2=LIM1+XBB4(SUB2)
      LIM6=LIM5+XBB4(SUB1)
      IF (LIM2.LT.LIM6) GO TO 170
      XDOSET(SUB1)=XDOSET(SUB1)+LIM2-LIM6+.00000001
  170 LIM3=XROSET(SUB2)-XBB3(SUB2)/2.
      LIM7=XROSET(SUB1)-XBB3(SUB1)/2.
      IF (LIM3.GT.LIM7) GO TO 180
      XROSET(SUB1)=XROSET(SUB1)-LIM7+LIM3-.00000001
      GO TO 190
  180 LIM4=LIM3+XBB3(SUB2)
      LIM8=LIM7+XBB3(SUB1)
      IF (LIM4.LT.LIM8) GO TO 190
      XROSET(SUB1)=XROSET(SUB1)+LIM4-LIM8+.00000001
```

```
190 CONTINUE                                                           INPU181   037630
    IF (AAA(2).NE.10HSYMMETRY  ) GO TO 210                             INPU182   037640
    DO 200 J=1,NBS                                                     INPU183   037650
    SUB1=(J-1)*25+I                                                    INPU184   037660
    LA(SUB1)=2.*LA(SUB1)                                               INPU185   037670
    KA(SUB1)=2.*KA(SUB1)                                               INPU186   037680
    XCOSET(SUB1)=0.                                                    INPU187   037690
200 XBB4(SUB1)=XBB4(SUB1)*2.                                           INPU188   037700
210 RETURN                                                             INPU189   037710
C                                                                      INPU190   037720
220 FORMAT (12F5.1)                                                    INPU191   037730
230 FORMAT (/32H THE DAMAGE FUNCTION ON FILE HAS,I3,50H REGIONS OF DIFINPU192   037740
   1FERENT PD. THE DATA CARDS FOR HOB =,F6.2,20H METERS SPECIFY THAT,IINPU193   037750
   23,9H BE USED.)                                                     INPU194   037760
240 FORMAT (23H0THIS IS A FATAL ERROR.)                                INPU195   037770
    END                                                                INPU196-  037780
```

```
      SUBROUTINE CMBN                                                   *3769*25 037790
C                                                                       CMBN  2  037800
      COMMON /CCM1/COM1(20100)                                          CMBN  3  037810
      DIMENSION ZZ(25,25,25),SZZ(50,25)                                 CMBN  4  037820
      EQUIVALENCE (COM1(1),ZZ(1,1,1)),(COM1(15626),SZZ(1,1))            CMBN  5  037830
C                                                                       CMBN  6  037840
      COMMON /COM3/COM3(10000)                                          CMBN  7  037850
      DIMENSION NN(25),HT(25),DF(25),HOB(25),PKO1(25),PKO12(25),PK1(25) CMBN  8  037860
     1          ,PK2(25),PK1PL(25),PK2PL(25),PK3PL(25),PKPS(25)         CMBN  9  037870
     2          ,PSALL(25),SZBAR(25),S1ZZ(25),S2ZZ(25),I1VEH(25)        CMBN 10  037880
     3          ,I2VEH(25),IFMT(1)                                      CMBN 11  037890
      EQUIVALENCE(COM3(1878),NN(1)),(COM3(1916),HT(1))                  CMBN 12  037900
     1          ,(COM3(3045),DF(1)),(COM3(3070),HOB(1))                 CMBN 13  037910
     2          ,(COM3(3095),PKO1(1)),(COM3(3120),PKO12(1))             CMBN 14  037920
     3          ,(COM3(3145),PK1(1)),(COM3(3170),PK2(1))                CMBN 15  037930
     4          ,(COM3(3195),PK1PL(1)),(COM3(3220),PK2PL(1))            CMBN 16  037940
     5          ,(COM3(3245),PK3PL(1)),(COM3(3270),PKPS(1))             CMBN 17  037950
     6          ,(COM3(3295),PSALL(1)),(COM3(3320),SZBAR(1))            CMBN 18  037960
     7          ,(COM3(3345),S1ZZ(1)),(COM3(3370),S2ZZ(1))              CMBN 19  037970
     8          ,(COM3(3395),I1VEH(1)),(COM3(3420),I2VEH(1))            CMBN 20  037980
     9          ,(COM3(3445),IFMT(1)),(COM3(9986),NVEH),(COM3(9987),NVS)CMBN 21  037990
     A          ,(COM3(10000),NHTS)                                     CMBN 22  038000
C                                                                       CMBN 23  038010
      COMMON /AAA/AAA(8)                                                CMBN 24  038020
      COMMON /PAGE/LINES,PAGE                                           CMBN 25  038030
C                                                                       CMBN 26  038040
      EQUIVALENCE (NHOB,FNHOB),(I,FI)                                   CMBN 27  038050
C                                                                       CMBN 28  038060
      DIMENSION FMT2(15),FMT4(17)                                       CMBN 29  038070
C                                                                       CMBN 30  038080
      INTEGER AAA,FMT2,FMT4,PAGE                                        CMBN 31  038090
C                                                                       CMBN 32  038100
      DATA (FMT2(I),I=1,15)/10H(19H      ,10HVOLLEY SIZ,10HE =,I5,9I1   CMBN 33D 038110
     1      ,10H1/)       ,10H(19H       ,10HVOLLEY SIZ,10HE =,I5,9I1   CMBN 34D 038120
     2      ,10H1/1H ,16X,,10H10I11/)   ,10H(19H       ,10HVOLLEY SIZ   CMBN 35D 038130
     3      ,10HE =,I5,9I1,10H1/1H ,16X,,10H10I11/1H ,,10H20X,5I11/)/   CMBN 36D 038140
      DATA (FMT4(I),I=1,17)/10H(14HOTARGE,10HT NUMBER,1,10H3,2H =,F9.   CMBN 37D 038150
     1      ,10H6,9F11.6) ,10H(14HOTARGE,10HT NUMBER,1,10H3,2H =,F9.    CMBN 38D 038160
     2      ,10H6,9F11.6/1,10HH ,20X,10F,10H11.6)     ,10H(14HOTARGE    CMBN 39D 038170
     3      ,10HT NUMBER,I,10H3,2H =,F9.,10H6,9F11.6/1,10HH ,20X,10F    CMBN 40D 038180
     4      ,10H11.6/1H ,2,10H4X,5F11.6)/                               CMBN 41D 038190
C                                                                       CMBN 42  038200
      IF (AAA(1).NE.10HOPTIMIZE  ) GO TO 20                             CMBN 43  038210
      AAA(6)=AAA(1)                                                     CMBN 44  038220
      AAA(7)=AAA(2)                                                     CMBN 45  038230
   10 AAA(8)=10HDONE                                                    CMBN 46  038240
      GO TO 50                                                          CMBN 47  038250
   20 IF (AAA(1).NE.10HALL HOBS  ) GO TO 40                             CMBN 48  038260
      DF(1)=1./FLOAT(NHTS)                                              CMBN 49  038270
      NHOB=NHTS                                                         CMBN 50  038280
      DO 30 I=1,NHOB                                                    CMBN 51  038290
      HOB(I)=HT(I)                                                      CMBN 52  038300
   30 DF(I)=DF(1)                                                       CMBN 53  038310
      GO TO 50                                                          CMBN 54  038320
   40 IF (AAA(1).NE.10HHOB DISTRI) GO TO 50                             CMBN 55  038330
      DECODE (10,520,AAA(3) )FNHOB                                      CMBN 56* 038340
      NHOB=FNHOB+.1                                                     CMBN 57  038350
      READ (5,520) (HOB(L),L=1,NHOB)                                    CMBN 58R 038360
      READ (5,520) (DF(L),L=1,NHOB)                                     CMBN 59R 038370
   50 DO 60 K=1,NVEH                                                    CMBN 60  038380
```

```
      DO 60 I=1,NVS                                                     CMBN 61   038390
   60 SZZ(K,I)=0.                                                       CMBN 62   038400
      WRITE (6,580)                                                     CMBN 63W  038410
      IF (AAA(6).EQ.10HOPTIMIZE  .AND.AAA(8).EQ.10HDONE      ) GO TO 180CMBN 64   038420
      LINES=(PAGE-21-4*((NHOB-1)/10)-(NVEH+1)*((NVS-1)/10+2))/2         CMBN 65   038430
      IF (LINES) 90,70,80                                               CMBN 66   038440
   70 WRITE (6,590)                                                     CMBN 67W  038450
      GO TO 90                                                          CMBN 68   038460
   80 ENCODE (10,610,IFMT(1) )LINES                                     CMBN 69*  038470
      WRITE (6,IFMT)                                                    CMBN 70W  038480
   90 WRITE (6,530)                                                     CMBN 71W  038490
      I1=1                                                              CMBN 72   038500
      I2=NHOB                                                           CMBN 73   038510
      IF (I2.GT.10) I2=10                                               CMBN 74   038520
  100 WRITE (6,540) (HOB(I),I=I1,I2)                                    CMBN 75W  038530
      WRITE (6,550) (DF(I),I=I1,I2)                                     CMBN 76W  038540
      IF (I2.EQ.NHOB) GO TO 110                                         CMBN 77   038550
      I1=I1+10                                                          CMBN 78   038560
      I2=I2+10                                                          CMBN 79   038570
      IF (I2.GT.NHOB) I2=NHOB                                           CMBN 80   038580
      GO TO 100                                                         CMBN 81   038590
  110 DO 140 L1=1,NHOB                                                  CMBN 82   038600
      DO 130 L2=1,NHTS                                                  CMBN 83   038610
      IF (HOB(L1).LT..999*HT(L2).OR.HOB(L1).GT.1.001*HT(L2)) GO TO 130  CMBN 84   038620
      DO 120 K=1,NVEH                                                   CMBN 85   038630
      DO 120 I=1,NVS                                                    CMBN 86   038640
  120 SZZ(K,I)=SZZ(K,I)+DF(L1)*ZZ(K,I,L2)                               CMBN 87   038650
      GO TO 140                                                         CMBN 88   038660
  130 CONTINUE                                                          CMBN 89   038670
      WRITE (6,560) L1,HOB(L1)                                          CMBN 90W  038680
      GO TO 510                                                         CMBN 91   038690
  140 CONTINUE                                                          CMBN 92   038700
      WRITE (6,570)                                                     CMBN 93W  038710
  150 IFMT2=1                                                           CMBN 94   038720
      IFMT4=1                                                           CMBN 95   038730
      IF (NVS.LT.11) GO TO 160                                          CMBN 96   038740
      IFMT2=5                                                           CMBN 97   038750
      IFMT4=5                                                           CMBN 98   038760
      IF (NVS.LT.21) GO TO 160                                          CMBN 99   038770
      IFMT2=10                                                          CMBN100   038780
      IFMT4=11                                                          CMBN101   038790
  160 WRITE (6,FMT2(IFMT2)) (NN(I),I=1,NVS)                             CMBN102W  038800
      DO 170 K=1,NVEH                                                   CMBN103   038810
      WRITE (6,FMT4(IFMT4)) K,(SZZ(K,I),I=1,NVS)                        CMBN104W  038820
  170 CONTINUE                                                          CMBN105   038830
      GO TO 300                                                         CMBN106   038840
C                                                                       CMBN107   038850
C     OPTIMIZATION                                                      CMBN108   038860
  180 DECODE (10,520,AAA(7) )FI                                         CMBN109*  038870
      I=FI+.1                                                           CMBN110   038880
      IF (I.EQ.0) GO TO 200                                             CMBN111   038890
      DO 190 I1=1,NVS                                                   CMBN112   038900
      IF (I.EQ.NN(I1)) GO TO 210                                        CMBN113   038910
  190 CONTINUE                                                          CMBN114   038920
  200 I1=1                                                              CMBN115   038930
  210 IF (NVEH.GE.NHOB) GO TO 230                                       CMBN116   038940
      L=NVEH+1                                                          CMBN117   038950
      DO 220 K=L,NHOB                                                   CMBN118   038960
      DO 220 I=1,NVS                                                    CMBN119   038970
  220 SZZ(K,I)=0.                                                       CMBN120   038980
```

```
  230 DO 240 L=1,NHTS                                                          CMBN121  038990
      DO 240 K=1,NVEH                                                          CMBN122  039000
C     PREVIOUSLY, WE HAD SZZ(NVEH,NVS) BUT NOW WE TREAT IT AS                  CMBN123  039010
C     SZZ(NHOB,NVS)                                                            CMBN124  039020
  240 SZZ(L,I1)=SZZ(L,I1)+ZZ(K,I1,L)                                           CMBN125  039030
      L1=1                                                                     CMBN126  039040
      DO 250 L=2,NHTS                                                          CMBN127  039050
      IF (SZZ(L,I1).GT.SZZ(L1,I1)) L1=L                                        CMBN128  039060
  250 CONTINUE                                                                 CMBN129  039070
C .   OPTIMUM HOB FOR THE SPECIFIED ATTACK LEVEL HAS BEEN DETERMINED.          CMBN130  039080
C     RETURN TO SZZ(NVEH,NVS)                                                  CMBN131  039090
      DO 260 K=1,NVEH                                                          CMBN132  039100
      DO 260 I=1,NVS                                                           CMBN133  039110
  260 SZZ(K,I)=ZZ(K,I,L1)                                                      CMBN134  039120
      LINES=(PAGE-6-((NVS-1)/10+2)*(NVEH+1))/2                                 CMBN135  039130
      IF (LINES) 290,270,280                                                   CMBN136  039140
  270 WRITE (6,590)                                                            CMBN137W 039150
      GO TO 290                                                                CMBN138  039160
  280 ENCODE (10,610,IFMT(1) )LINES                                            CMBN139* 039170
      WRITE (6,IFMT)                                                           CMBN140W 039180
  290 WRITE (6,600) HT(L1),NN(I1)                                              CMBN141W 039190
      GO TO 150                                                                CMBN142  039200
C                                                                              CMBN143  039210
  300 LINES=(3*NVEH-1)*NVEH+4                                                  CMBN144  039220
      IF (AAA(4).NE.10HSYMMETRIC ) GO TO 330                                   CMBN145  039230
      K=2*NVEH                                                                 CMBN146  039240
      LINES=LINES+3                                                            CMBN147  039250
      DO 320 I=1,NVEH                                                          CMBN148  039260
      DO 310 J=1,NVS                                                           CMBN149  039270
  310 SZZ(K,J)=SZZ(I,J)                                                        CMBN150  039280
  320 K=K-1                                                                    CMBN151  039290
      NVEH=2*NVEH                                                              CMBN152  039300
C                                                                              CMBN153  039310
  330 IF (LINES.GT.14) LINES=14                                                CMBN154  039320
      DO 360 J=1,NVS                                                           CMBN155  039330
      PSALL(J)=1.                                                              CMBN156  039340
      SUM=0.                                                                   CMBN157  039350
      SUM2=0.                                                                  CMBN158  039360
      DO 340 I=1,NVEH                                                          CMBN159  039370
      PS=1.-SZZ(I,J)                                                           CMBN160  039380
      PSALL(J)=PSALL(J)*PS                                                     CMBN161  039390
      PKPS(I)=9999999.                                                         CMBN162  039400
      IF (PS.GT..0000001) PKPS(I)=SZZ(I,J)/PS                                  CMBN163  039410
  340 SUM=SUM+PKPS(I)                                                          CMBN164  039420
      PK1(J)=PSALL(J)*SUM                                                      CMBN165  039430
      DO 350 L=1,NVEH                                                          CMBN166  039440
      DO 350 M=1,NVEH                                                          CMBN167  039450
      IF (M.GT.L) SUM2=SUM2+PKPS(L)*PKPS(M)                                    CMBN168  039460
  350 CONTINUE                                                                 CMBN169  039470
      PK2(J)=PSALL(J)*SUM2                                                     CMBN170  039480
      PK1PL(J)=1.-PSALL(J)                                                     CMBN171  039490
      PK2PL(J)=PK1PL(J)-PK1(J)                                                 CMBN172  039500
      PK3PL(J)=PK2PL(J)-PK2(J)                                                 CMBN173  039510
      PK01(J)=PSALL(J)+PK1(J)                                                  CMBN174  039520
  360 PK012(J)=PK01(J)+PK2(J)                                                  CMBN175  039530
C                                                                              CMBN176  039540
      WRITE (6,580)                                                            CMBN177W 039550
      LINES=(PAGE-8-((NVS-1)/10+2)*LINES)/2                                    CMBN178  039560
      IF (LINES) 390,380,370                                                   CMBN179  039570
  370 ENCODE (10,610,IFMT(1) )LINES                                            CMBN180* 039580
```

```
      WRITE (6,IFMT)                                                    CMBN181W 039590
      GO TO 390                                                         CMBN182  039600
  380 WRITE (6,590)                                                     CMBN183W 039610
  390 IF ((NVS-1)/10-1) 430,410,400                                     CMBN184  039620
  400 I3=20                                                             CMBN185  039630
      I4=I3+1                                                           CMBN186  039640
      GO TO 420                                                         CMBN187  039650
  410 I3=NVS                                                            CMBN188  039660
  420 I1=10                                                             CMBN189  039670
      I2=I1+1                                                           CMBN190  039680
      GO TO 440                                                         CMBN191  039690
  430 I1=NVS                                                            CMBN192  039700
C                                                                       CMBN193  039710
  440 IF (AAA(4).NE.10HSYMMETRIC ) WRITE (6,620)                        CMBN194W 039720
      IF (AAA(4).EQ.10HSYMMETRIC ) WRITE (6,630)                        CMBN195W 039730
      WRITE (6,640) (NN(I),I=1,I1)                                      CMBN196W 039740
      IF (NVS.GT.10) WRITE (6,770) (NN(I),I=I2,I3)                      CMBN197W 039750
      IF (NVS.GT.20) WRITE (6,790) (NN(I),I=I4,NVS)                     CMBN198W 039760
      WRITE (6,650) (PSALL(I),I=1,I1)                                   CMBN199W 039770
      IF (NVS.GT.10) WRITE (6,780) (PSALL(I),I=I2,I3)                   CMBN200W 039780
      IF (NVS.GT.20) WRITE (6,800) (PSALL(I),I=I4,NVS)                  CMBN201W 039790
      IF (NVEH.LT.2) GD TO 450                                          CMBN202  039800
      WRITE (6,660) (PK1(I),I=1,I1)                                     CMBN203W 039810
      IF (NVS.GT.10) WRITE (6,780) (PK1(I),I=I2,I3)                     CMBN204W 039820
      IF (NVS.GT.20) WRITE (6,800) (PK1(I),I=I4,NVS)                    CMBN205W 039830
      IF (NVEH.LT.3) GO TO 450                                          CMBN206  039840
      WRITE (6,670) (PK2(I),I=1,I1)                                     CMBN207W 039850
      IF (NVS.GT.10) WRITE (6,760) (PK2(I),I=I2,I3)                     CMBN208W 039860
      IF (NVS.GT.20) WRITE (6,800) (PK2(I),I=I4,NVS)                    CMBN209W 039870
  450 WRITE (6,680) (PK1PL(I),I=1,I1)                                   CMBN210W 039880
      IF (NVS.GT.10) WRITE (6,780) (PK1PL(I),I=I2,I3)                   CMBN211W 039890
      IF (NVS.GT.20) WRITE (6,800) (PK1PL(I),I=I4,NVS)                  CMBN212W 039900
      IF (NVEH.LT.2) GO TO 510                                          CMBN213  039910
      WRITE (6,690) (PK2PL(I),I=1,I1)                                   CMBN214W 039920
      IF (NVS.GT.10) WRITE (6,780) (PK2PL(I),I=I2,I3)                   CMBN215W 039930
      IF (NVS.GT.20) WRITE (6,800) (PK2PL(I),I=I4,NVS)                  CMBN216W 039940
      IF (NVEH.LT.3) GO TO 460                                          CMBN217  039950
      WRITE (6,700) (PK3PL(I),I=1,I1)                                   CMBN218W 039960
      IF (NVS.GT.10) WRITE (6,780) (PK3PL(I),I=I2,I3)                   CMBN219W 039970
      IF (NVS.GT.20) WRITE (6,800) (PK3PL(I),I=I4,NVS)                  CMBN220W 039980
  460 WRITE (6,710) (PKO1(I),I=1,I1)                                    CMBN221W 039990
      IF (NVS.GT.10) WRITE (6,780) (PKO1(I),I=I2,I3)                    CMBN222W 040000
      IF (NVS.GT.20) WRITE (6,800) (PKO1(I),I=I4,NVS)                   CMBN223W 040010
      IF (NVEH.LT.3) GO TO 510                                          CMBN224  040020
      WRITE (6,720) (PKO12(I),I=1,I1)                                   CMBN225W 040030
      IF (NVS.GT.10) WRITE (6,780) (PKO12(I),I=I2,I3)                   CMBN226W 040040
      IF (NVS.GT.20) WRITE (6,800) (PKO12(I),I=I4,NVS)                  CMBN227W 040050
C                                                                       CMBN228  040060
      IF (AAA(4).EQ.10HSYMMETRIC ) NVEH=NVEH/2                          CMBN229  040070
      IF (NVEH.LT.2) GO TO 510                                          CMBN230  040080
      DO 480 J=1,NVS                                                    CMBN231  040090
      SZBAR(J)=0.                                                       CMBN232  040100
      DO 470 I=1,NVEH                                                   CMBN233  040110
  470 SZBAR(J)=SZBAR(J)+SZZ(I,J)                                        CMBN234  040120
  480 SZBAR(J)=SZBAR(J)/FLOAT(NVEH)                                     CMBN235  040130
      WRITE (6,730) (SZBAR(J),J=1,I1)                                   CMBN236W 040140
      IF (NVS.GT.10) WRITE (6,780) (SZBAR(J),J=I2,I3)                   CMBN237W 040150
      IF (NVS.GT.20) WRITE (6,800) (SZBAR(J),J=I4,NVS)                  CMBN236W 040160
      IF (NVEH.LT.3) GO TO 510                                          CMBN239  040170
      DO 490 J=1,NVS                                                    CMBN240  040180
```

```
      S1ZZ(J)=SZZ(1,J)                                                  CMBN241  040190
      I1VEH(J)=1                                                        CMBN242  040200
      DO 490 I=2,NVEH                                                   CMBN243  040210
      IF (SZZ(I,J).GE.S1ZZ(J)) GO TO 490                                CMBN244  040220
      S1ZZ(J)=SZZ(I,J)                                                  CMBN245  040230
      I1VEH(J)=I                                                        CMBN246  040240
  490 CONTINUE                                                          CMBN247  040250
      DO 500 J=1,NVS                                                    CMBN248  040260
      S2ZZ(J)=SZZ(1,J)                                                  CMBN249  040270
      I2VEH(J)=1                                                        CMBN250  040280
      DO 500 I=2,NVEH                                                   CMBN251  040290
      IF (SZZ(I,J).LE.S2ZZ(J)) GO TO 500                                CMBN252  040300
      S2ZZ(J)=SZZ(I,J)                                                  CMBN253  040310
      I2VEH(J)=I                                                        CMBN254  040320
  500 CONTINUE                                                          CMBN255  040330
      WRITE (6,740) (I1VEH(J),J=1,I1)                                   CMBN256W 040340
      IF (NVS.GT.10) WRITE (6,770) (I1VEH(J),J=I2,I3)                   CMBN257W 040350
      IF (NVS.GT.20) WRITE (6,790) (I1VEH(J),J=I4,NVS)                  CMBN258W 040360
      WRITE (6,760) (S1ZZ(J),J=1,I1)                                    CMBN259W 040370
      IF (NVS.GT.10) WRITE (6,780) (S1ZZ(J),J=I2,I3)                    CMBN260W 040380
      IF (NVS.GT.20) WRITE (6,800) (S1ZZ(J),J=I4,NVS)                   CMBN261W 040390
      WRITE (6,750) (I2VEH(J),J=1,I1)                                   CMBN262W 040400
      IF (NVS.GT.10) WRITE (6,770) (I2VEH(J),J=I2,I3)                   CMBN263W 040410
      IF (NVS.GT.20) WRITE (6,790) (I2VEH(J),J=I4,NVS)                  CMBN264W 040420
      WRITE (6,760) (S2ZZ(J),J=1,I1)                                    CMBN265W 040430
      IF (NVS.GT.10) WRITE (6,780) (S2ZZ(J),J=I2,I3)                    CMBN266W 040440
      IF (NVS.GT.20) WRITE (6,800) (S2ZZ(J),J=I4,NVS)                   CMBN267W 040450
C                                                                       CMBN268  040460
  510 IF (AAA(6).NE.10HOPTIMIZE  .OR.AAA(6).EQ.10HDONE      ) RETURN    CMBN269  040470
      GO TO 10                                                          CMBN270  040480
C                                                                       CMBN271  040490
  520 FORMAT (8F10.4)                                                   CMBN272  040500
  530 FORMAT (1H ,51X,28HTHE HOB DISTRIBUTION FOLLOWS////)              CMBN273  040510
  540 FORMAT (1H0,24X,6HHOB = ,10F10.2)                                 CMBN274  040520
  550 FORMAT (1H0,5X,25HFRACTION OF POPULATION = ,10F10.3)              CMBN275  040530
  560 FORMAT (/////5H0HOB(,I2,2H)=,F6.2,112H METERS, OF THIS DISTRIBUTIOCMBN276  040540
     1N, DOES NOT COME WITHIN .1 PER CENT OF ANY HOB FOR WHICH PROBABILICMBN277  040550
     2TIES OF DAMAGE ARE/68HOCURRENTLY AVAILABLE. THEREFOR, THIS HOB DISCMBN278  040560
     3TRIBUTION IS TERMINATED.)                                         CMBN279  040570
  570 FORMAT (/////1H0,25X,81HPROBABILITIES OF DAMAGE AVERAGED OVER THE CMBN280  040580
     1HOB DISTRIBUTION FOR EACH TARGET FOLLOW////)                      CMBN281  040590
  580 FORMAT (1H1)                                                      CMBN282  040600
  590 FORMAT (1H )                                                      CMBN283  040610
  600 FORMAT (1H ,18X,46HPD'S FOR EACH TARGET WITH OPTIMUM FUZING (HOB=,CMBN284  040620
     1F6.2,22H METERS, ATTACK LEVEL=,I3,19H ROUNDS PER CANNON)////)     CMBN285  040630
  610 FORMAT (1H(,I2,7H(/))   )                                         CMBN286  040640
  620 FORMAT (1H ,46X,38HTHE SYSTEM DAMAGE PROBABILITIES FOLLOW,6(/))   CMBN287  040650
  630 FORMAT (1H ,16X,65HTHE SYSTEM (INCLUDING TARGET REFLECTIONS DUE TOCMBN288  040660
     1 SYMMETRY) DAMAGE PROBABILITIES FOLLOW,6(/))                      CMBN289  040670
  640 FORMAT (26H            VOLLEY SIZE =,I7,9I10)                     CMBN290  040680
  650 FORMAT (/26H0            ALL SURVIVE =,10F10.5)                   CMBN291  040690
  660 FORMAT (26H0     EXACTLY ONE DAMAGED =,10F10.5)                   CMBN292  040700
  670 FORMAT (26H0     EXACTLY TWO DAMAGED =,10F10.5)                   CMBN293  040710
  680 FORMAT (26H0    AT LEAST ONE DAMAGED =,10F10.5)                   CMBN294  040720
  690 FORMAT (26H0    AT LEAST TWO DAMAGED =,10F10.5)                   CMBN295  040730
  700 FORMAT (26H0 AT LEAST THREE DAMAGED =,10F10.5)                    CMBN296  040740
  710 FORMAT (26H0  LESS THAN TWO DAMAGED =,10F10.5)                    CMBN297  040750
  720 FORMAT (26H0LESS THAN THREE DAMAGED =,10F10.5)                    CMBN298  040760
  730 FORMAT (26H0      AVERAGE PER TARGET =,10F10.5)                   CMBN299  040770
  740 FORMAT (26H0  LEAST PROBABLE TARGET =,10I10)                      CMBN300  040780
```

```
750 FORMAT (26H0    MOST PROBABLE TARGET =,10I10)                         CMBN301   040790
760 FORMAT (26H0            PROBABILITY =,10F10.5)                        CMBN302   040800
770 FORMAT (1H ,25X,10I10)                                                CMBN303   040810
780 FORMAT (1H ,28X,10F10.5)                                              CMBN304   040820
790 FORMAT (1H ,26X,10I10)                                                CMBN305   040830
800 FORMAT (1H ,31X,9F10.5)                                               CMBN306   040840
    END                                                                   CMBN307-  040850
```

```
      SUBROUTINE SEXUN4
C
      COMMON /COM3/COM3(10000)
      DIMENSION VDYN(92),NSP(46),M(40,46),Q(4C,46),SGST(46),PLATE(20)
     1          ,T(20),THETA(20),METAL(20),CV(20),ALFAV(20),BETAV(20)
     2          ,GAMAV(20),LAMDAV(20),KFCTRV(20),CM(20),ALFAM(20)
     3          ,BETAM(20),GAMAM(20),LAMDAM(20),KFCTRM(20),EXPVR(20)
     4          ,EXPMR(2C),QVR(20),QMR(20),LMDAV(20),LMDAM(20),GMAV(20)
     5          ,GMAM(20)
      EQUIVALENCE  (COM3(231),VDYN(1)),(COM3(323),NSP(1))
     1            ,(COM3(369),M(1,1)),(COM3(2209),Q(1,1))
     2            ,(COM3(4049),SGST(1)),(COM3(4167),PLATE(1))
     3            ,(COM3(4207),T(1)),(COM3(4227),THETA(1))
     4            ,(COM3(4247),METAL(1)),(COM3(4267),CV(1))
     5            ,(COM3(4287),ALFAV(1)),(COM3(4307),BETAV(1))
     6            ,(COM3(4327),GAMAV(1)),(COM3(4347),LAMDAV(1))
     7            ,(COM3(4367),KFCTRV(1)),(COM3(4387),CM(1))
     8            ,(COM3(4407),ALFAM(1)),(COM3(4427),BETAM(1))
     9            ,(COM3(4447),GAMAM(1)),(COM3(4467),LAMDAM(1))
     A            ,(COM3(4487),KFCTRM(1)),(COM3(4507),EXPVR(1))
     B            ,(COM3(4527),EXPMR(1)),(COM3(4547),QVR(1))
     C            ,(COM3(4567),QMR(1)),(COM3(4587),LMDAV(1))
     D            ,(COM3(4607),LMDAM(1)),(COM3(4627),GMAV(1))
     E            ,(COM3(4647),GMAM(1)),(COM3(4667),RATIOM)
      EQUIVALENCE  (COM3(4668),RATIOV),(COM3(4669),RATOM1)
     1            ,(COM3(4670),RATOV1),(COM3(9910),PHK)
     2            ,(COM3(9911),MMMR1),(COM3(9912),VVVR1)
     3            ,(COM3(9913),MMMR),(COM3(9914),VVVR),(COM3(9915),K1)
     4            ,(COM3(9916),L),(COM3(9917),K2),(COM3(9918),KEY)
     5            ,(COM3(9919),SECANT),(COM3(9920),SGSTBR)
     6            ,(COM3(9921),PHKBAR),(COM3(9922),ITHETA)
     7            ,(COM3(9923),NTHETA,FNTHET),(COM3(9924),QQQ)
     8            ,(COM3(9925),ITHOR),(COM3(9926),NTHOR,FNTHOR)
     9            ,(COM3(9927),IPLATE),(COM3(9928),NPLATE,FNPLAT)
     A            ,(COM3(9934),VCTFLG),(COM3(9937),FRGFLG)
     B            ,(COM3(9940),SHAPE),(COM3(9941),AA),(COM3(9942),NNSP)
     C            ,(COM3(9961),VTERM),(COM3(9983),K22),(COM3(9984),K222)
C
      COMMON /AAA/AAA(8)
      COMMON /CONSTT/CONSTT(6)
      EQUIVALENCE (CONSTT(3),DGTOR)
      COMMON /PAGE/LINES,PAGE
C
      DIMENSION IFMT1(23),WORD(2)
      EQUIVALENCE (DUMMY,IDUMMY)
C
      INTEGER AAA,FRGFLG,PAGE,PLATE,VCTFLG,WORD
      REAL KFCTRM,KFCTRV,LAMDAM,LAMDAV,LMDAM,LMDAV,M,MMMR,MMMR1
C
      DATA (WORD(I),I=1,2)  /10H (MED)    ,10H (MAX)    /
      DATA (IFMT1(I),I=1,23)/10H(1H1,3I(/),1CH,1H ,42X,4,10H6HTHE THOR
     1      ,1CH CONSTANTS,10H FOR THE F,10HOLLOWING C,10HASES ARE,5
     2      ,10H(/),1H ,55,10HX,1HC,7X,5,10HHALPHA,5X,,10H4HBETA,6X,
     3      ,10H5HGAMMA,4X,10H,6HLAMBDA,,10H5X,5HKFCTR,10H(//1H0,23X
     4      ,10H ,A1C,5X,1,10HOHV RESIDU,10HAL,6F10.3/,10H1HC,38X,10
     5      ,10HHM RESIDUA,10HL,6F10.3)),10H(    1H1/ ,10H 1H ,42X,4/
C
   10 READ (5,280) (AAA(I),I=1,8)
      IF (AAA(1).EQ.10HCOMPUTE   .OR.AAA(1).EQ.10H COMPUTE  .OR.AAA(1).E
     1Q.10H  COMPUTE .OR.AAA(1).EQ.10H   COMPUTE) GO TO 100
```

```
C                                                                            SEXU461  041460
      IF (AAA(1).NE.1CHFRAGMENTAT) GO TO 20                                  SEXU462  041470
      FRGFLG=1                                                               SEXU463  041480
      VCTFLG=1                                                               SEXU464  041490
      CALL FRAG1 (1HO)                                                       SEXU465  041500
      WRITE (6,320) SHAPE,AA                                                 SEXU466W 041510
      CALL FRAG2 (1HO)                                                       SEXU467  041520
      GO TO 10                                                               SEXU468  041530
C                                                                            SEXU469  041540
   20 IF (AAA(1).NE.10HTERMINAL V) GO TO 30                                  SEXU470  041550
      VCTFLG=1                                                               SEXU471  041560
      DECODE (10,290,AAA(3) )VTERM                                           SEXU472* 041570
      GO TO 10                                                               SEXU473  041580
C                                                                            SEXU474  041590
   30 IF (AAA(1).NE.10HTHOR DATA .AND.AAA(1).NE.10H THOR DATA) GO TO 60 SEXU475  041600
      DECODE (10,290,AAA(2) )FNTHOR                                          SEXU476* 041610
      NTHOR=FNTHOR+.1                                                        SEXU477  041620
      READ (5,310) (METAL(I),CV(I),ALFAV(I),BETAV(I),GAMAV(I),LAMDAV(I),SEXU478R 041630
     1  KFCTRV(I),CM(I),ALFAM(I),BETAM(I),GAMAM(I),LAMDAM(I),KFCTRM(I),  SEXU479R 041640
     2  I=1,NTHOR)                                                           SEXU480R 041650
      LINES=PAGE-7-5*NTHOR                                                   SEXU481  041660
      IF (LINES.LT.0) GO TO 40                                               SEXU482  041670
      LINES=LINES/2+2                                                        SEXU483  041680
      ENCODE (20,330,IFMT1(1) )LINES                                         SEXU484* 041690
      GO TO 50                                                               SEXU485  041700
   40 IFMT1(1)=IFMT1(22)                                                     SEXU486  041710
      IFMT1(2)=IFMT1(23)                                                     SEXU487  041720
   50 WRITE (6,IFMT1) (METAL(I),CV(I),ALFAV(I),BETAV(I),GAMAV(I),           SEXU488W 041730
     1  LAMDAV(I),KFCTRV(I),CM(I),ALFAM(I),BETAM(I),GAMAM(I),LAMDAM(I),  SEXU489W 041740
     2  KFCTRM(I),I=1,NTHOR)                                                 SEXU490W 041750
      GO TO 10                                                               SEXU491  041760
C                                                                            SEXU492  041770
   60 IF (AAA(1).NE.10HANGLES    .AND.AAA(1).NE.10H ANGLES   .AND.AAA(1)SEXU493  041780
     1.NE.10H  ANGLES  .AND.AAA(1).NE.10H   ANGLES .AND.AAA(1).NE.10H    SEXU494  041790
     2 ANGLES) GO TO 70                                                      SEXU495  041800
      DECODE (70,290,AAA(2) )FNTHET,(THETA(I),I=1,6)                         SEXU496* 041810
      NTHETA=FNTHET+.1                                                       SEXU497  041820
      IF (NTHETA.GT.6) READ (5,290) (THETA(I),I=7,NTHETA)                    SEXU498R 041830
      GO TO 10                                                               SEXU499  041840
C                                                                            SEXU100  041850
   70 IF (AAA(1).NE.10HARMOR     .AND.AAA(1).NE.10H ARMOR    .AND.AAA(1)SEXU101  041860
     1.NE.10H  ARMOR   .AND.AAA(1).NE.10H   ARMOR  .AND.AAA(1).NE.10H    SEXU102  041870
     2 ARMOR .AND.AAA(1).NE.10H        AEMOR) GO TO 80                       SEXU103  041880
      DECODE (10,290,AAA(2) )FNPLAT                                          SEXU104* 041890
      NPLATE=FNPLAT+.1                                                       SEXU105  041900
      READ (5,300) (PLATE(IPLATE),T(IPLATE),IPLATE=1,NPLATE)                 SEXU106R 041910
      GO TO 10                                                               SEXU107  041920
C                                                                            SEXU108  041930
   80 IF (AAA(1).NE.10HINCAPACITA) GO TO 90                                  SEXU109  041940
      DUMMY=INCAP(DUMMY,DUMMY,AAA(4))                                        SEXU110  041950
      GO TO 10                                                               SEXU111  041960
C                                                                            SEXU112  041970
   90 IF (AAA(1).EQ.10HSECTION IV) GO TO 10                                  SEXU113  041980
      RETURN                                                                 SEXU114  041990
C                                                                            SEXU115  042000
C     END OF INPUT DATA                                                      SEXU116  042010
C                                                                            SEXU117  042020
  100 IF (FRGFLG.EQ.1) GO TO 130                                             SEXU118  042030
      FRGFLG=1                                                               SEXU119  042040
      DO 120 K2=K22,K222                                                     SEXU120  042050
```

```
      IF (SGST(K2).EQ.0.) GO TO 120
      NNSP=NSP(K2)
      DO 110 K1=1,NNSP
  110 M(K1,K2)=10.**M(K1,K2)
  120 CONTINUE
  130 IF (VCTFLG.NE.1) GO TO 140
      CALL VECTR
      VCTFLG=0
  140 QQQ=(.0610234*SHAPE)**(-2./3.)
      DUMMY=PPKFCN(QQQ,DUMMY,IDUMMY)
      IPLATE=0
  150 IPLATE=IPLATE+1
      DO 160 ITHOR=1,NTHOR
      IF (METAL(ITHOR).EQ.PLATE(IPLATE)) GO TO 190
  160 CONTINUE
      WRITE (6,380) IPLATE,PLATE(IPLATE)
      IF (IPLATE.EQ.NPLATE) GO TO 180
      IPLATE=IPLATE+1
      DO 170 JPLATE=IPLATE,NPLATE
      PLATE(JPLATE-1)=PLATE(JPLATE)
  170 T(JPLATE-1)=T(JPLATE)
      IPLATE=IPLATE-1
  180 NPLATE=NPLATE-1
      GO TO 200
  190 EXPVR(IPLATE)=2.*ALFAV(ITHOR)/3.+BETAV(ITHOR)
      EXPMR(IPLATE)=2.*ALFAM(ITHOR)/3.+BETAM(ITHOR)
      QQQ=T(IPLATE)*SHAPE**(-2./3.)
      QVR(IPLATE)=10.**CV(ITHOR)*QQQ**ALFAV(ITHOR)
      QMR(IPLATE)=10.**CM(ITHOR)*QQQ**ALFAM(ITHOR)
      LMDAV(IPLATE)=LAMDAV(ITHOR)
      LMDAM(IPLATE)=LAMDAM(ITHOR)
      GMAV(IPLATE)=GAMAV(ITHOR)
      GMAM(IPLATE)=GAMAM(ITHOR)
C     FROM THOR 7C, KFCTRM(2024)=.964 AND KFCTRV(2024)=.965
      RATIOM=1.
      RATIOV=1.
      IF (KFCTRM(IPLATE).NE.0.) RATIOM=KFCTRM(IPLATE)/.964
      IF (KFCTRV(IPLATE).NE.0.) RATIOV=KFCTRV(IPLATE)/.965
      RATOM1=1.-RATIOM
      RATOV1=1.-RATIOV
  200 IF (IPLATE.LT.NPLATE) GO TO 150
      DO 270 IPLATE=1,NPLATE
      DO 260 ITHETA=1,NTHETA
      WRITE (6,340) PLATE(IPLATE),T(IPLATE),THETA(ITHETA),VTERM
      PHKBAR=0.
      SGSTBR=0.
      SECANT=1000000000.
      IF (ABS(THETA(ITHETA)).LE.89.) SECANT=1./COS(THETA(ITHETA)*DGTOR)
      QVR(IPLATE)=QVR(IPLATE)*SECANT**GMAV(IPLATE)
      QMR(IPLATE)=QMR(IPLATE)*SECANT**GMAM(IPLATE)
      DO 240 K2=K22,K222
      IF (SGST(K2).EQ.0.) GO TO 240
      NNSP=NSP(K2)
      DO 230 L=1,NNSP
      KEY=2
      K1=NNSP-L+1
  210 VVVR=VDYN(K2+(KEY-1)*46)
      MMMR=M(K1,K2)
      VVVR1=VVVR-QVR(IPLATE)*MMMR**EXPVR(IPLATE)*VVVR**LMDAV(IPLATE)
      VVVR1=VVVR*RATOV1+VVVR1*RATIOV
```

```
      IF (VVVR1.LE.0.) GO TO (230,240), KEY                                  SEXU181   042660
      MMMR1=MMMR-QMR(IPLATE)*MMMR**EXPMR(IPLATE)*VVVR**LMDAM(IPLATE)         SEXU182   042670
      MMMR1=MMMR*RATOM1+MMMR1*RATIOM                                         SEXU183   042680
      IF (MMMR1.LE.0.) GO TO (240,220), KEY                                  SEXU184   042690
      MMMR=MMMR1                                                             SEXU185   042700
      VVVR=VVVR1                                                             SEXU186   042710
      IF (SGSTBR.EQ.0.) WRITE (6,350)                                        SEXU187W  042720
      PHK=PKFCN(MMMR,VVVR,IDUMMY)                                            SEXU188   042730
      WRITE (6,360) K2,VDYN(K2+(KEY-1)*46),WORD(KEY),M(K1,K2),VVVR,MMMR,     SEXU189W  042740
     1  PHK                                                                  SEXU190W  042750
      PHKBAR=PHKBAR+PHK*SGST(K2)*Q(K1,K2)                                    SEXU191   042760
      SGSTBR=SGSTBR+SGST(K2)*Q(K1,K2)                                        SEXU192   042770
  220 KEY=KEY-1                                                              SEXU193   042780
      IF (KEY.EQ.1) GO TO 210                                                SEXU194   042790
  230 CONTINUE                                                               SEXU195   042800
  240 CONTINUE                                                               SEXU196   042810
      IF (SGSTBR.NE.0.) GO TO 250                                            SEXU197   042820
      WRITE (6,390)                                                          SEXU198W  042830
      GO TO 270                                                              SEXU199   042840
  250 PHKBAR=PHKBAR/SGSTBR                                                   SEXU200   042850
      WRITE (6,370) PHKBAR                                                   SEXU201W  042860
      QVR(IPLATE)=QVR(IPLATE)*SECANT**(-GMAV(IPLATE))                        SEXU202   042870
      QMR(IPLATE)=QMR(IPLATE)*SECANT**(-GMAM(IPLATE))                        SEXU203   042880
  260 CONTINUE                                                               SEXU204   042890
  270 CONTINUE                                                               SEXU205   042900
      GO TO 10                                                               SEXU206   042910
C                                                                            SEXU207   042920
  280 FORMAT (8A10)                                                          SEXU208   042930
  290 FORMAT (8E10.4)                                                        SEXU209   042940
  300 FORMAT (4(A10,F10.4))                                                  SEXU210   042950
  310 FORMAT (A10,6F10.4/10X,6F10.4)                                         SEXU211   042960
  320 FORMAT (1H1/15H SHAPE FACTOR =,F6.1,5H  A =,F5.3/)                     SEXU212   042970
  330 FORMAT (5H(1H1,,I2,13H(/),1H ,42X,4)                                   SEXU213   042980
  340 FORMAT (1H1,12X,A10,F7.2,25H INCHES THICK     THETA =,F6.2,34H DEGSEXU214   042990
     1REES      PROJECTILE VELOCITY =,F8.2,16H FEET PER SECOND)              SEXU215   043000
  350 FORMAT (/1H0,28X,5HSPRAY,6X,6HV IMPACT,9X,8HM IMPACT,5X,10HV RESIDSEXU216   043010
     1UAL,4X,10HM RESIDUAL,5X,4HPI/H)                                        SEXU217   043020
  360 FORMAT (1H0,28X,I3,E16.4,A6,3E14.4,F9.4)                               SEXU218   043030
  370 FORMAT (15H0   MEAN PI/H =,E12.4)                                      SEXU219   043040
  380 FORMAT (1H1,24(/),1H ,35X,12HPLATE NUMBER,I3,4H IS ,A10,1H./1H0,35SEXU220   043050
     1X,61HTHOR CONSTANTS FOR THIS MATERIAL ARE NOT CURRENTLY AVAILABLE.SEXU221   043060
     2/1H0,35X,35HTHEREFOR, THIS PLATE IS TERMINATED.)                       SEXU222   043070
  390 FORMAT (1H ,10(/),1H0,43X,45HTHIS PLATE ASSEMBLY SUSTAINED NO PERFSEXU223   043080
     1ORATIONS)                                                              SEXU224   043090
      END                                                                    SEXU225-  043100
```

```
      FUNCTION PKFCN (AA,BB,II)                                         *4301*27 043110
C                                                                       PKFCN 2  043120
      A=AA                                                              PKFCN 3  043130
      B=BB                                                              PKFCN 4  043140
      PKFCN=0.                                                          PKFCN 5  043150
      V50=                                                              PKFCN 6  043160
      IF (B.LT.V50) GO TO 10                                            PKFCN 7  043170
      C=                                                                PKFCN 8  043180
      IF (C.GT.COEF2) PKFCN=                                            PKFCN 9  043190
   10 RETURN                                                            PKFCN10  043200
C                                                                       PKFCN11  043210
      ENTRY INCAP                                                       PKFCN12E 043220
      DECODE (40,30,II) KEY,COEF1,COEF2,COEF3                           PKFCN13* 043230
      WRITE (6,40) KEY,COEF1,COEF2,COEF3                                PKFCN14W 043240
      GO TO 10                                                          PKFCN15  043250
C                                                                       PKFCN16  043260
      ENTRY PPKFCN                                                      PKFCN17E 043270
      D=AA                                                              PKFCN18  043280
      IF (KEY.EQ.10HSUMMER    ) GO TO 20                                PKFCN19  043290
      COEF4=14630000.*D                                                 PKFCN20  043300
      COEF5=-1666300.*D                                                 PKFCN21  043310
      GO TO 10                                                          PKFCN22  043320
   20 COEF4=7543000.*D                                                  PKFCN23  043330
      COEF5=-1794200.*D                                                 PKFCN24  043340
      GO TO 10                                                          PKFCN25  043350
C                                                                       PKFCN26  043360
   30 FORMAT (A10,7E10.4)                                               PKFCN27  043370
   40 FORMAT (1H1,29(/),1H ,37X,57HTHE INCAPACITATION PARAMETERS FOR THEPKFCN28  043380
     1 FOLLOWING CASES ARE//1H0,37X,A10,3E17.5)                         PKFCN29  043390
      END                                                               PKFCN30- 043400
```

The Following Page is Blank.

APPENDIX B

PROBABILITY OF INCAPACITATION EQUATIONS

This appendix presents sufficient information concerning the probability of incapacitation equations for soldiers (references 5 and 8) to allow those equations to be encoded in a manner compatible with the LASVEM computer program. The equations themselves have been deleted from this report in order to keep it unclassified.

Cards PKFCN 6 with cards PKFCN 20, PKFCN 21, PKFCN 23, and PKFCN 24 compute a V_{50} level as described in paragraph 1 of reference 8. That reference uses a factor, 'A', the mean presented area in centimeters squared. Centimeters must be converted to inches and the presented area must be eliminated:

$$A = \left(\frac{M}{.0610234 \cdot SHAPE}\right)^{2/3} \qquad \text{Equation B-1}$$

Note that card SEXU129 incorporates the factor, .0610234, and that card SEXU188, which is the call to function PKFCN, defines AA, BB and, in turn A, B, as fragment mass and speed, respectively.

Cards PKFCN 8 and PKFCN 9 compute the probability of incapacitating a soldier with a fragment of mass A (in grains) and velocity B (in fps). Reference 6 uses the three variables a, b, and n which are encoded as COEF1, COEF2, and COEF3, respectively.

The following page is blank.

APPENDIX C

AN ILLUSTRATIVE EXAMPLE

This appendix provides an example of the use of this methodology from start to finish. Although the example is typical, it is very simple. None of the data necessarily represent actual or developmental systems. Not all the options of the program are exercised, nor is all of the output shown.

We wish to determine measures of the damage sustained by a force of four APC's when the force is fired upon by artillery firing HE shell fuzed for air burst. The APC's are oriented in the direction of fire. A scenario such as this might be useful for determining the protection provided to the occupants by the APC armor against friendly supporting fires. The scenario is depicted in Figure C-1.

Various numbers of rounds will be fired by one howitzer battery consisting of six cannon. The aiming policy is battery parallel sheaf with a hypothetical cannon at the battery's geometrical center aiming at the center of the APC force. Ballistic data and fuze data are shown in Table C-1. APC structure is indicated by a shop diagram at Figure C-2. Data for the individual APC armor plates are in Table C-2. We use one perforation as the damage criterion.

First, program section IV will be used to determine which plates have any chance of sustaining perforations. There are two reasons for using section IV:

1. Plates identified as not sustaining any perforations can be eliminated from consideration, thus conserving both machine time and user effort.

2. An estimate of the average probability of incapacitation of soldiers hit by residual fragments will be produced.

The input data deck is at Figures C-3 and C-4. Plates D and E are not considered at all because they are shielded from fragmentation by suspension and mobility components and from direct hits by the rest of the APC. Data from other sources indicates that the spaced armor will not sustain perforations. Results of this first step are summarized in Table C-3. A sample of the output is at Figure C-5.

Plates C, F, and H sustain no damage from fragmentation; however, they may sustain massive damage from direct hits and very near misses (see paragraph 4.1). Although we may process all three of these plates for direct hits and very near misses without error, we need process only plate F, because the hits and very near misses on plate C

Text continues on page 170.

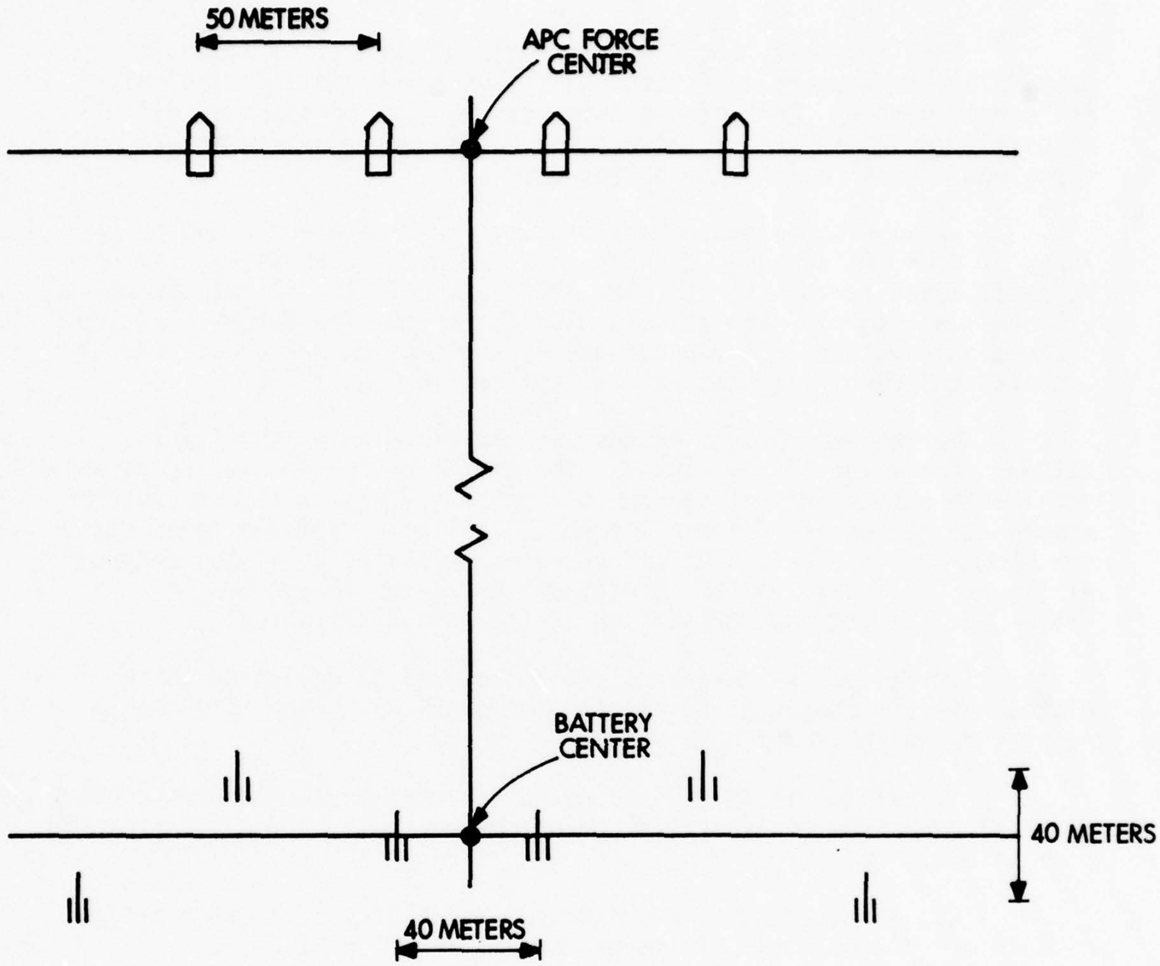


Figure C-1. (U) Geometry of the Firing Battery and APC Force.

TABLE C-1. BALLISTIC AND FUZE DATA

Projectile Terminal Velocity	850. Ft/Sec		
Projectile Angle of Fall	30.°		
Fuze Reliability	.95		
Fuze HOB Distribution:			
HOB (Meters):	.3	5.	12.
Fraction of Population:	.03	.80	.17
Precision Error Standard Deviation, Range	36. meters		
Precision Error Standard Deviation, Deflection	6. meters		
MPI Error Standard Deviation, Range	45. meters		
MPI Error Standard Deviation, Deflection	27. meters		

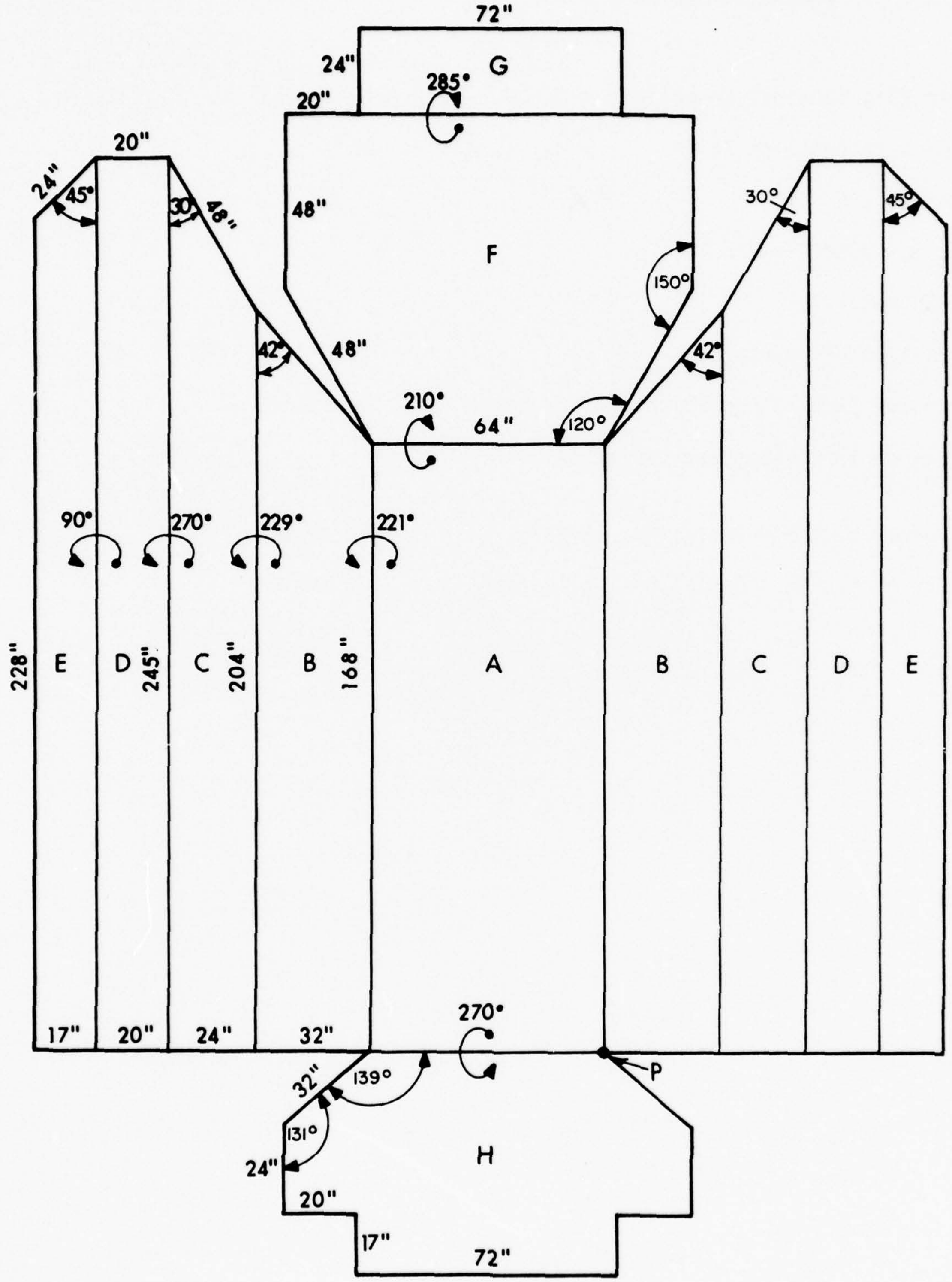


Figure C-2. (U) APC Structure, Shop Diagram.

TABLE C-2. PLATE DATA

Plate or Plate Pair	Material	Thickness in Inches	Inclination to Horizontal	Surface Area in $Inches^2$	Height of Top Edge or Surface above Ground in Inches
A	7039 Aluminum	1.5	0.°	10752.	78.
B	2024 Aluminum	1.6	41.°	2X5948.	78.
C	Spaced Armor	*	90.°	2X5389.	57.
D, E	Shielded from fragmentation by tracks, roadwheels, skirts, etc.				
F	Hard Homogeneous Steel	2.0	30.°	9034.	78.
G	7039 Aluminum	2.0	135.°	1728.	33.
H	Hard Homogeneous Steel	2.0	90.°	5772.	78.

*The spaced armor consists of 3 plates. From outermost to innermost, they are:

.25 inches hard homogeneous steel
.20 inches hard homogeneous steel
1.12 inches 7039 aluminum

Next page is blank.

FRAGMENTATION DATA		640.	0.032				
19.	15.	13.	5.	4.	1.	3.	5.
7.	16.	21.	26.	14.	7.	6.	7.
10.	13.	22.	24.				
1.	0.0645	0.6800	0.0968	1.4900	0.2581	3.4680	0.0323
5.2600	0.0323	9.9500	0.0968	10.5330	0.0323	21.0200	0.0968
132.6670	0.0645	157.0000	0.0323	245.0000	0.0323	259.0000	0.0323
309.0000	0.0323	493.0000	0.0323	622.0000	0.0645	5250.0000	
2.	0.2857	0.4210	0.2381	1.5540	0.0952	2.9150	0.0238
5.8900	0.0238	9.2300	0.0238	11.7700	0.0476	18.5900	0.0238
22.2200	0.0238	25.4700	0.0714	40.6730	0.0476	66.5000	0.0476
480.0000	0.0476	735.5000					
3.	0.1667	0.4500	0.1667	1.5000	0.1667	20.7000	0.3333
39.5300	0.1667	112.0000					
4.	0.7500	0.2730	0.0833	2.9000	0.0833	9.7400	0.0833
27.2500							
5.	1.0000	0.0000					
6.	0.3333	2.5900	0.3333	7.3600	0.3333	637.0000	
7.	0.4000	0.3120	0.2000	2.8250	0.2000	6.3200	0.1000
65.2000	0.1000	72.4000					
8.	0.1667	0.6100	0.1667	1.4450	0.2500	17.5000	0.1667
39.3000	0.0833	56.5200	0.0833	492.0000	0.0833	1649.0000	
9.	0.1379	0.6750	0.1034	1.5800	0.1379	2.6350	0.0345
8.7200	0.0345	12.7000	0.0345	31.4400	0.0345	64.8000	0.0690
76.0000	0.0690	93.0000	0.0345	128.0000	0.0345	295.0000	0.1034
364.0000	0.0345	424.0000	0.0690	666.5000	0.0345	969.0000	0.0345
1352.0000							
10.	0.0976	0.7280	0.0488	1.3900	0.0488	2.9600	0.0488
6.4750	0.0732	12.7200	0.0488	17.2300	0.0488	22.3000	0.0488
26.6000	0.0244	51.5000	0.0976	64.0000	0.0244	79.3400	0.0488
83.5000	0.0244	98.9000	0.0244	101.0000	0.0244	197.0000	0.0244
221.0000	0.0488	357.5000	0.0488	441.5000	0.0732	543.6670	0.0488
952.0000	0.0244	1535.0000					
11.	0.0274	0.6300	0.0137	1.1900	0.0685	3.2860	0.0548
6.4350	0.0137	8.1700	0.0411	12.4330	0.0822	16.8620	0.0137
20.1500	0.0411	32.2670	0.0548	44.6200	0.0274	55.0000	0.0411
76.3330	0.0137	84.7200	0.0822	112.0000	0.0137	128.0000	0.0822
175.3330	0.0548	274.5000	0.0548	333.2500	0.0411	426.0000	0.0411
651.3330	0.0685	844.4000	0.0137	1525.0000	0.0137	1634.0000	0.0137
2135.0000	0.0137	2634.0000	0.0137	3050.0000			
12.	0.1250	0.7000	0.0625	1.7100	0.0625	6.8400	0.0625
9.8400	0.0625	11.9600	0.0625	19.1400	0.0625	28.3700	0.0625
66.0000	0.0625	75.8000	0.0625	96.8000	0.1250	175.5000	0.0625
1200.0000	0.0625	1470.0000	0.0625	2760.0000			
13.	0.3636	3.5850	0.1818	14.2700	0.0909	18.7900	0.0909
32.7200	0.0909	64.2200	0.0909	71.4000	0.0909	128.0000	
14.	0.4091	0.4440	0.3636	1.1390	0.0909	3.2900	0.0455
52.0000	0.0455	85.0000	0.0455	149.0000			
15.	0.7000	0.3730	0.1000	1.4470	0.1000	2.6120	0.0167
5.1400	0.0167	11.0400	0.0333	21.6350	0.0333	32.3200	
16.	0.4167	0.2000	0.0833	1.4500	0.0833	3.0950	0.0417
[illegible]	0.0417	16.5700	0.0417	22.8000	0.1667	46.1000	0.0417

5.1400	0.0167	11.0400	0.0333	21.6350	0.0333	32.3200	
16.	0.4167	0.2000	0.0833	1.4500	0.0833	3.0950	0.0417
9.9700	0.0417	16.5700	0.0417	22.8000	0.1667	46.1000	0.0417
56.8000	0.0417	73.0600	0.0417	84.4400			
17.	0.5000	0.2600	0.0526	1.8000	0.1053	3.3920	0.0526
12.3100	0.0263	40.8000	0.0526	64.4000	0.0263	88.2600	0.0526
128.5000	0.0263	159.0000	0.0263	417.0000	0.0263	803.0000	0.0263
1200.0000	0.0263	1455.0000					
18.	0.1274	0.4920	0.1863	1.3780	0.1765	3.4610	0.0686
6.2430	0.0098	8.4200	0.0392	13.2150	0.0196	15.3600	0.0294
21.1600	0.0392	27.5380	0.0392	42.2500	0.0294	59.7270	0.0098
65.6000	0.0294	74.9530	0.0490	113.4000	0.0196	141.5000	0.0686
173.7140	0.0098	210.0000	0.0098	351.0000	0.0098	427.0000	0.0098
539.0000	0.0098	1300.0000	0.0098	1768.0000			
19.	0.0534	0.5000	0.0840	1.4360	0.1374	3.5540	0.0667
6.4190	0.0382	9.3120	0.0840	11.8020	0.0382	17.0320	0.0076
20.9000	0.0687	30.9840	0.0611	40.3350	0.0534	55.5170	0.0153
62.7000	0.0229	75.4070	0.0382	83.5440	0.0229	94.5400	0.0305
110.5000	0.0153	132.0000	0.0305	159.7500	0.0153	270.0000	0.0153
346.5000	0.0153	450.5000	0.0153	656.0000	0.0305	814.0000	0.0382
1871.0000							
1.	0.0	5.0	2300.0	2452.0	0.0	0.0	
2.	5.0	15.0	2550.0	3014.0	0.0	0.0	
3.	15.0	25.0	2730.0	3088.0	0.0	0.0	
4.	25.0	35.0	2900.0	3163.0	0.0	0.0	
5.	35.0	45.0	3038.0	3238.0	0.0	0.0	
6.	45.0	55.0	3250.0	3950.0	33.6	40.0	
7.	55.0	65.0	2900.0	3550.0	111.0	117.0	
8.	65.0	75.0	3650.0	4200.0	132.8	129.0	
9.	75.0	85.0	3950.0	4500.0	322.5	299.0	
10.	85.0	95.0	4200.0	4700.0	455.6	416.0	
11.	95.0	105.0	3150.0	4450.0	811.1	752.0	
12.	105.0	115.0	1850.0	2250.0	177.0	172.0	
13.	115.0	125.0	1950.0	2050.0	121.4	128.0	
14.	125.0	135.0	2400.0	3250.0	243.3	290.0	
15.	135.0	145.0	2350.0	2950.0	545.6	775.0	
16.	145.0	155.0	2050.0	2400.0	150.6	275.0	
17.	155.0	165.0	2200.0	2550.0	147.6	394.0	
18.	165.0	175.0	2250.0	2800.0	135.2	711.0	
19.	175.0	180.0	2550.0	2900.0	65.5	2740.0	

FIGURE C-3. INPUT DATA, FRAGMENTATION

The Following Page is Blank.

```
SECTION IV
          *** THE FRAGMENTATION DATA DECK OF FIGURE C-3 IS INSERTED HERE ***
TERMINAL VELOCITY    850.
THOR DATA 3.
AL 7039    7.047     1.029     -1.072    1.251     -.139     1.016
           -6.663    .227      .694      -.361     1.901     1.021
AL 2024    7.047     1.029     -1.072    1.251     -.139     .965
           -6.663    .227      .694      -.361     1.901     .964
HHS        6.475     .889      -.945     1.262     .019
           -2.264    .346      .629      .327      .880
ANGLES     1.        0.
INCAPACITATION PARAMETERS      WINTER    .0010454  31000.    .48781
ARMOR      4.
AL 7039    1.5       AL 7039   2.0       AL 2024   1.6       HHS       2.0
COMPUTE
END OF DATA
```

FIGURE C-4. INPUT DATA DECK FOR PROGRAM SECTION IV

TABLE C-3. RESULTS OF PROGRAM SECTION IV

PLATE	PERFORATED	$P_{I/H}$
A	yes	.83
B	yes	.62
C	no	-
D	no	-
E	no	-
F	no	-
G	yes	.65
H	no	-

AL 7039 1.50 INCHES THICK THETA = 0.00 DEGREES PROJECTILE VELOCITY = 850.00 FEET PER SECOND

SPRAY	V IMPACT	M IMPACT	V RESIDUAL	M RESIDUAL	PI/H
8	.4561E+04 (MAX)	.1649E+04	.9909E+03	.1225E+04	.9946
8	.4021E+04 (MED)	.1649E+04	.3875E+03	.1315E+04	.9339
9	.4722E+04 (MAX)	.1352E+04	.8863E+03	.9686E+03	.9863
9	.4182E+04 (MED)	.1352E+04	.2808E+03	.1048E+04	0.0000
9	.4722E+04 (MAX)	.9690E+03	.3599E+03	.6797E+03	.8446
10	.4776E+04 (MAX)	.1535E+04	.1129E+04	.1099E+04	.9957
10	.4285E+04 (MED)	.1535E+04	.5828E+03	.1180E+04	.9690
10	.4776E+04 (MAX)	.9520E+03	.3908E+03	.6608E+03	.8579
11	.4383E+04 (MAX)	.3050E+04	.1552E+04	.2388E+04	1.0000
11	.3117E+04 (MED)	.3050E+04	.1480E+03	.2704E+04	0.0000
11	.4383E+04 (MAX)	.2634E+04	.1387E+04	.2049E+04	.9998
11	.4383E+04 (MAX)	.2135E+04	.1134E+04	.1645E+04	.9987
11	.4383E+04 (MAX)	.1634E+04	.7803E+03	.1244E+04	.9879
11	.4383E+04 (MAX)	.1525E+04	.6830E+03	.1157E+04	.9791

MEAN PI/H = .8274E+00

FIGURE C-5. OUTPUT OF PROGRAM SECTION IV FOR ONE OF THE ARMOR PLATES.

and on plate H will be accounted for on plates B and F and on plate A, respectively. Thus, for HOB = .3 meters, (HOB assumed to be half the projectile length for impact functioning rounds) we consider plates A, B, F, and G, and for HOB = 5. and 12. meters, we consider plates A, B, and G. It should be obvious that plate G will sustain little damage.

For the computations of program sections I and II, we use a four inch cell at HOB = .3 meters and an eight inch cell otherwise. We assign a value of four feet to the blast effects sphere radius. We define plate A to be the reference plate to which the other plates will be assembled; therefore, the offsets of the various plates' central cells are defined with respect to the central cell of plate A (central cells and their offsets are discussed in paragraphs 4.1 and 5.1). Data for the individual plates of the structure serving as the model of our APC are at Table C-4. The input data deck for this second stage of the computations is at Figure C-6.

Samples of the output from program section I are at Figures C-7 through C-9. Table C-5 contains the important parameters computed by program section I.

We note that our intuition concerning plate G was correct; it has sustained no damage. We note further that for HOB's greater than 473. inches (55 eight inch cells plus 33 inches height of the top edge of plate G above the ground), plate G would present zero surface area to the burst and so could not sustain any perforations. Considerations such as these can be important when machine time becomes large. The source of the uncertainty in the values for plate B at HOB = 12. meters is clear from examination of Figure C-9; perforations might be sustained beyond the left deflection limit of the matrix used. Any error, however, is quite small. Finally, all these values refer to a half-infinite ground plane.

The output of program section II is at Figures C-10 through C-12. We note that, due to the possible error in the values for plate B at HOB = 12. meters, the 12 meter values for the complete structure may incorporate a small error. Again these values refer to a half-infinite ground plane.

Figures C-13 and C-14 present an index of the expected perforations file and a tabulation of the damage function file, respectively. Although not produced by program section II, these are presented here because the input data card which caused them to be produced was a part of the data deck shown in Figure C-6. The first several logical records (which are not shown) of the damage function file pertain to the individual plates with non-zero probability of damage (Table C-5). Damage function file logical records 8, 9, and 10 refer to the complete structure at HOB = .3, 5., and 12. meters, respectively. Had the data deck of Figure C-6 been split up into two parts, one for program section I and the other for program section II, then an examination of the section I

Text continues on page 193.

TABLE C-4. DATA FOR STRUCTURE APPROXIMATING APC OF FIGURE C-2.

Plate	Dimensions in No. of Cells: Along Range Axis	Dimensions in No. of Cells: Along Deflection Axis	Dimensions in No. of Cells: Along Slope	Inclination to Horizontal in Degrees	Height of Top Surface or Edge Above Ground	Surface Area Scale Factor	Central Cell Offset: Along Range Axis	Central Cell Offset: Along Deflection Axis
				4-inch cells				
A	42	16		0	78	1.000	22†	9†
B	46	16*	8	41	78	.9899	2	0
F		26	22	30	78	1.013	21	0
G		18	6	135	33	1.000	40	0
				8-inch cells				
A	21	8		0	78	1.000	11†	5†
B	23	8*	4	41	78	.9899	1	0
G		9	3	135	33	1.000	21	0

*Separation between top edges of two halves of plate pair.

†Location of central cell, rather than offset, with respect to point P of Figure C-2.

Next page is blank.

```
BASIC DATA              4.        100.      100.      55.       0.
1.        .3
1.        1.5
1.
42.       16.                                                   78.
1.        1.
THOR DATA
AL 7039   7.047     1.029     -1.072    1.251     -.139     1.016
          -6.663    .227      .694      -.361     1.901     1.021
30.       850.      48.
       *** THE FRAGMENTATION DATA DECK OF FIGURE C-3 IS INSERTED HERE ***
CONTOUR             NO PROBABILITY MATRIX
BASIC DATA          4.        100.      100.      55.       0.
1.        .3
1.        2.
1.
          18.       6.        1.        135.                33.
1.        1.
30.       850.      48.
CONTOUR             NO PROBABILITY MATRIX
BASIC DATA          4.        100.      100.      55.       0.
1.        .3
1.        1.6
1.
46.       16.       8.        2.                  41.       78.       .9899
1.        1.
THOR DATA
AL 2024   7.047     1.029     -1.072    1.251     -.139     .965
          -6.663    .227      .694      -.361     1.901     .964
30.       850.      48.
CONTOUR             NO PROBABILITY MATRIX
BASIC DATA          4.        100.      100.      55.       0.
1.        .3
1.        2.
1.
          26.       22.       1.        30.                 78.       1.013
1.        1.
THOR DATA
HHS       6.475     .889      -.945     1.262     .019
          -2.264    .346      .629      .327      .880
30.       850.      48.
CONTOUR             NO PROBABILITY MATRIX
BASIC DATA          8.        100.      100.      55.       0.
2.        5.        12.
1.        1.5
1.
21.       8.                                                  78.
1.        1.
```

```
1.
21.        8.                                                 78.
1.         1.
THOR DATA
AL 7C39    7.047     1.029     -1.072    1.251     -.139     1.016
           -6.663    .227      .694      -.361     1.901     1.021
30.        850.      48.
CONTOUR              NO PROBABILITY MATRIX
BASIC DATA           8.        100.      100.      55.       0.
2.         5.        12.
1.         2.
1.
           9.        3.        1.        135.                33.
1.         1.
30.        850.      48.
CONTOUR              NO PROBABILITY MATRIX
BASIC DATA           8.        1CC.      100.      55.       0.
2.         5.        12.
1.         1.6
1.
23.        8.        4.        2.                  41.       78.       .9899
1.         1.
THOR DATA
AL 2024    7.047     1.029     -1.072    1.251     -.139     .965
           -6.663    .227      .694      -.361     1.901     .964
30.        850.      48.
CONTOUR              NO PROBABILITY MATRIX
ASSEMBLE             3.        CONTOUR             NO PROBABILITY MATRIX
1.         1.
4.         1.        3.        4.        2.
SHIFT X =            2.        21.       40.
3.         5.        9.        7.
SHIFT X =            1.        21.
3.         6.        10.       8.
EXPECTED PERFORATIONS FILE               DAMAGE FUNCTION FILE
END OF DATA
```

FIGURE C-6. INPUT DATA DECK FOR PROGRAM SECTIONS I AND II

The following page is blank.

```
PROBABILITY OF SUSTAINING AT LEAST  1 PERFORATIONS

HOB=  .30 METERS    THICKNESS=1.50 INCHES    CELL= 4.00 INCHES    NX= 42 CELLS    NY= 16 CELLS    NZ=  0 CELLS    T=  78.0 INCHES

SCALE FACTOR APPLIED TO INPUT PLATE = 1./1.000

NXX=100 CELLS    NYY=100 CELLS    PROJECTILE LOCATION WAS (  55.5,     .5)

THE EXPECTED PERFORATIONS MATRIX FOR THIS CASE IS ON THE EXPECTED PERFORATIONS FILE (PMAT348    ) AS RECORD NUMBER    1

LETHAL AREA= 13.626 SQUARE METERS    AVERAGE PROBABILITY OF DAMAGE =1.000 OVER   13.626 SQUARE METERS


                         OUTPUT OF THE PROBABILITY OF DAMAGE MATRIX HAS BEEN SUPPRESSED


THE PROBABILITY CONTOURS FOR THIS TARGET, HOB, AND DAMAGE CRITERION FOLLOW. THE NUMBERS ARE PD*10. (TRUNCATED), AND E=.0005, *=1.0

      222222233333333334444444444555555555566666666667777777777888888888
      345678901234567890123456789012345678901234567890123456789012345678

 20   ******************************************************************   20
 19   *                                                                *   19
 18   *                                                                *   18
 17   *                                                                *   17
 16   *                                                                *   16
 15   *                                                                *   15
 14   *                                                                *   14
 13   *                                                                *   13
 12   *                                                                *   12
 11   *                                                                *   11
 10   *                                                                *   10
  9   *                                                                *    9
  8   *                                                                *    8
  7   *                                                                *    7
  6   *                                                                *    6
  5   *                                                                *    5
  4   *                                                                *    4
  3   *                                                                *    3
  2   *                                                                *    2
  1   ******************************************************************    1

      222222233333333334444444444555555555566666666667777777777888888888
      345678901234567890123456789012345678901234567890123456789012345678


THESE DATA ARE ON THE DAMAGE FUNCTION FILE (DFCN348    ) AS RECORD NUMBER   1


                  AVERAGE PD WITHIN     LETHAL AREA IN METERS**2     DAMAGE AREA IN METERS**2
                     EACH REGION
                                        PER REGION   CUMULATIVE     PER REGION   CUMULATIVE
```

THESE DATA ARE ON THE DAMAGE FUNCTION FILE (DFCN348) AS RECORD NUMBER 1

AVERAGE PD WITHIN EACH REGION	LETHAL AREA IN METERS**2		DAMAGE AREA IN METERS**2	
	PER REGION	CUMULATIVE	PER REGION	CUMULATIVE
1.000	13.626	13.626	13.626	13.626

AVERAGE PD WITHIN EACH REGION	RECTANGULARIZED, UNNORMALIZED RANGE DATA				RECTANGULARIZED, UNNORMALIZED DEFLECTION DATA			
	EXTENT OF REGION		LOCATION OF REGION CENTER WRT PROJECTILE		EXTENT OF REGION		LOCATION OF REGION CENTER WRT PROJECTILE	
	METERS	CELLS	METERS	CELLS	METERS	CELLS	METERS	CELLS
1.000	6.706	66.0	0.000	0.0	2.032	20.0	1.016	10.0

FIGURE C-7. SOME OUTPUT OF PROGRAM SECTION I FOR HOB = .3 METERS.

The following page is blank.

PROBABILITY OF SUSTAINING AT LEAST 1 PERFORATIONS

HOB= 5.00 METERS   THICKNESS=1.60 INCHES   CELL= 8.00 INCHES   NX= 23 CELLS   NY= 8 CELLS   NZ= 4 CELLS   T= 78.0 INCHES

SCALE FACTOR APPLIED TO INPUT PLATE = 1./ .990

ALPHA=   0.0 DEGREES   BETA= 41.0 DEGREES   2 SIDES

NXX=100 CELLS   NYY=100 CELLS   PROJECTILE LOCATION WAS (55.5,   .5)

THE EXPECTED PERFORATIONS MATRIX FOR THIS CASE IS ON THE EXPECTED PERFORATIONS FILE (PMAT348   ) AS RECORD NUMBER   9

LETHAL AREA= 21.695 SQUARE METERS   AVERAGE PROBABILITY OF DAMAGE = .226 OVER   95.835 SQUARE METERS

OUTPUT OF THE PROBABILITY OF DAMAGE MATRIX HAS BEEN SUPPRESSED

THE PROBABILITY CONTOURS FOR THIS TARGET, HOB, AND DAMAGE CRITERION FOLLOW. THE NUMBERS ARE PD*10. (TRUNCATED), AND E=.C005, *=1.0

```
    3333344444444445555555555666666666677777777
    5678901234567890123456789012345678901234567

76  0000EEEEEEEEEEEEEEEEEEEEEEEEEE0000C0C000000   76
75  C E                          E            0   75
74  E                            E            0   74
73  E                            E            0   73
72  E                            E            0   72
71  E                            E            0   71
70  E                            E            0   70
69  E                             E           0   69
68  E                             E           0   68
67  E                            E            0   67
66  E                            E            0   66
65  E                            E            C   65
64  E                            E            0   64
63  E                            E            0   63
62  E                            E            0   62
61  E                            E            0   61
60  E                            E            0   60
59  E                            E            0   59
58  E                           E             0   58
57  E     1               1     E             C   57
56  E    1                 1    E             0   56
55  E    1                 1    E             0   55
54  E    1                 1    E             0   54
53  E   1                  1    E             0   53
52  E   1                   1   E             0   52
51  E   1                  1    E             0   51
50  E   1                   1   E             0   50
49  E   1                   1   E             0   49
48  E  1                    1  E              0   48
47  E  1                    1  E              0   47
46  E  1                    1  E              0   46
45  E  1                     1      E         0   45
44  E 1  2                 2 1       E        0   44
43  E 1 2                  2 1       E        0   43
42  E 1 2                   2 1      E        0   42
41  E 1 2                   2 1       E       0   41
40  E 12      3           3 2 1      E        0   40
39  E 12     3            3 2 1      E        0   39
38  E1 2   3               3 21      E        C   38
37  E1 2   3               3 2 1            E 0   37
36  E1 2  3                3 2  1           E 0   36
35  E1 2 3             4  4 32    1          EC   35
34  E1  23       4         43 2    1         E0   34
33  E12 3      4           4 3 2     1       E0   33
```

```
40 E 12      3           3 2 1      E        0 40
39 E 12     3            3 2 1      E        0 39
38 E1 2   3               3 21      E        0 38
37 E1 2   3               3 2 1            E 0 37
36 E1 2  3                3 2  1           E 0 36
35 E1 2 3             4  4 32    1          EC 35
34 E1 23        4         43 2    1         E0 34
33 E12 3      4           4 3 2     1       E0 33
32 E12 3     4             43    2    1      E 32
31 E12 3     4       5   5 4 3     2    1    E 31
30 E123    4      5       5 4 3    2    1    E 30
29 E123   4      5        5 4  3    2   1   E0 29
28 E123  4    5            54   3   2   1   E0 28
27 E123 4   5      6      654    3  2   1   E0 27
26 E13 4   5     6        65 4   3  2   1  E 0 26
25 E234   5    6          6 5 4  3  2  1   E 0 25
24 E234  5   6             65  4 3  2  1  E  0 24
23 E234 5  6      7      7 65  4 3  2  1  E  0 23
22 E234 5 6      7        765  4 3 2  1  E   0 22
21 E234 5  6     7       7 65  43  2  1 E    0 21
20 E24  5 6     7         765  4 3 2  1 E    0 20
19 E24  5 6    7          765  43  2 1 E     0 19
18 E24  5 6    7         7 65  43  2 1 E     0 18
17 E234 5  6    7        7 65  43  2 1E      0 17
16 E234  5  6             65  4 3  21 E      0 16
15 E234  5  6             65  4 3 2 1E       0 15
14 E234   5   6            65 4  3 2 1E       0 14
13 E23 4   5  6          6 54  3  21 E       0 13
12 E13    4  5            54 3   2 1E        0 12
11 E12      3             32      1 E        0 11
10 OE1      2             21        E        0 10
 9 OE                      E                 0  9
 8 OOEEEEEEEEEEEEEEEEEEEEEEE000000000000000000  8

   3333344444444445555555555666666666677777777
   5678901234567890123456789012345678901234567
```

THESE DATA ARE ON THE DAMAGE FUNCTION FILE (DFCN348) AS RECORD NUMBER 6

AVERAGE PD WITHIN EACH REGION	LETHAL AREA IN METERS**2		DAMAGE AREA IN METERS**2	
	PER REGION	CUMULATIVE	PER REGION	CUMULATIVE
.720	2.110	2.110	2.932	2.932
.653	4.126	6.236	6.317	9.249
.549	2.720	8.956	4.955	14.204
.448	3.055	12.012	6.813	21.017
.348	3.175	15.186	9.125	30.142
.251	2.472	17.659	9.868	40.010
.143	2.410	20.069	16.888	56.898
.042	1.627	21.695	38.937	95.835

AVERAGE PD WITHIN EACH REGION	RECTANGULARIZED, UNNORMALIZED RANGE DATA				RECTANGULARIZED, UNNORMALIZED DEFLECTION DATA			
	EXTENT OF REGION		LOCATION OF REGION CENTER WRT PROJECTILE		EXTENT OF REGION		LOCATION OF REGION CENTER WRT PROJECTILE	
	METERS	CELLS	METERS	CELLS	METERS	CELLS	METERS	CELLS
.720	2.438	12.0	-.610	-3.0	1.422	7.0	3.962	19.5
.653	3.658	18.0	-1.016	-5.0	3.048	15.0	3.962	19.5
.549	4.267	21.0	-1.118	-5.5	4.064	20.0	4.267	21.0
.448	5.486	27.0	-1.118	-5.5	4.877	24.0	4.674	23.0
.348	5.893	29.0	-.914	-4.5	6.096	30.0	5.080	25.0

The following page is bla

.251	2.472	17.659	9.868	40.010
.143	2.410	20.069	16.888	56.898
.042	1.627	21.695	38.937	95.835

AVERAGE PD WITHIN EACH REGION	RECTANGULARIZED, UNNORMALIZED RANGE DATA				RECTANGULARIZED, UNNORMALIZED DEFLECTION DATA			
	EXTENT OF REGION		LOCATION OF REGION CENTER WRT PROJECTILE		EXTENT OF REGION		LOCATION OF REGION CENTER WRT PROJECTILE	
	METERS	CELLS	METERS	CELLS	METERS	CELLS	METERS	CELLS
.720	2.438	12.0	-.610	-3.0	1.422	7.0	3.962	19.5
.653	3.658	18.0	-1.016	-5.0	3.048	15.0	3.962	19.5
.549	4.267	21.0	-1.118	-5.5	4.064	20.0	4.267	21.0
.448	5.486	27.0	-1.118	-5.5	4.877	24.0	4.674	23.0
.348	5.893	29.0	-.914	-4.5	6.096	30.0	5.080	25.0
.251	6.706	33.0	-.711	-3.5	7.112	35.0	5.385	26.5
.143	7.518	37.0	-.305	-1.5	9.754	48.0	6.706	33.0
.042	8.738	43.0	.102	.5	14.021	69.0	8.433	41.5

3

FIGURE C-8. SOME OUTPUT OF PROGRAM SECTION I FOR HOB = 5 METERS.

The following page is blank.

PROBABILITY OF SUSTAINING AT LEAST 1 PERFORATIONS

HOB=12.00 METERS   THICKNESS=1.60 INCHES   CELL= 8.00 INCHES   NX= 23 CELLS   NY= 8 CELLS   NZ= 4 CELLS   T= 78.0 INCHES

SCALE FACTOR APPLIED TO INPUT PLATE = 1./ .990

ALPHA=   0.0 DEGREES   BETA= 41.0 DEGREES   2 SIDES

NXX=100 CELLS   NYY=100 CELLS   PROJECTILE LOCATION WAS (55.5,   .5)

THE TARGET MIGHT SUSTAIN ADDITIONAL PERFORATIONS WERE ITS CENTER FURTHER REMOVED FROM THE PROJECTILE THAN ROW   100

THE EXPECTED PERFORATIONS MATRIX FOR THIS CASE IS ON THE EXPECTED PERFORATIONS FILE (PMAT348   ) AS RECORD NUMBER   10

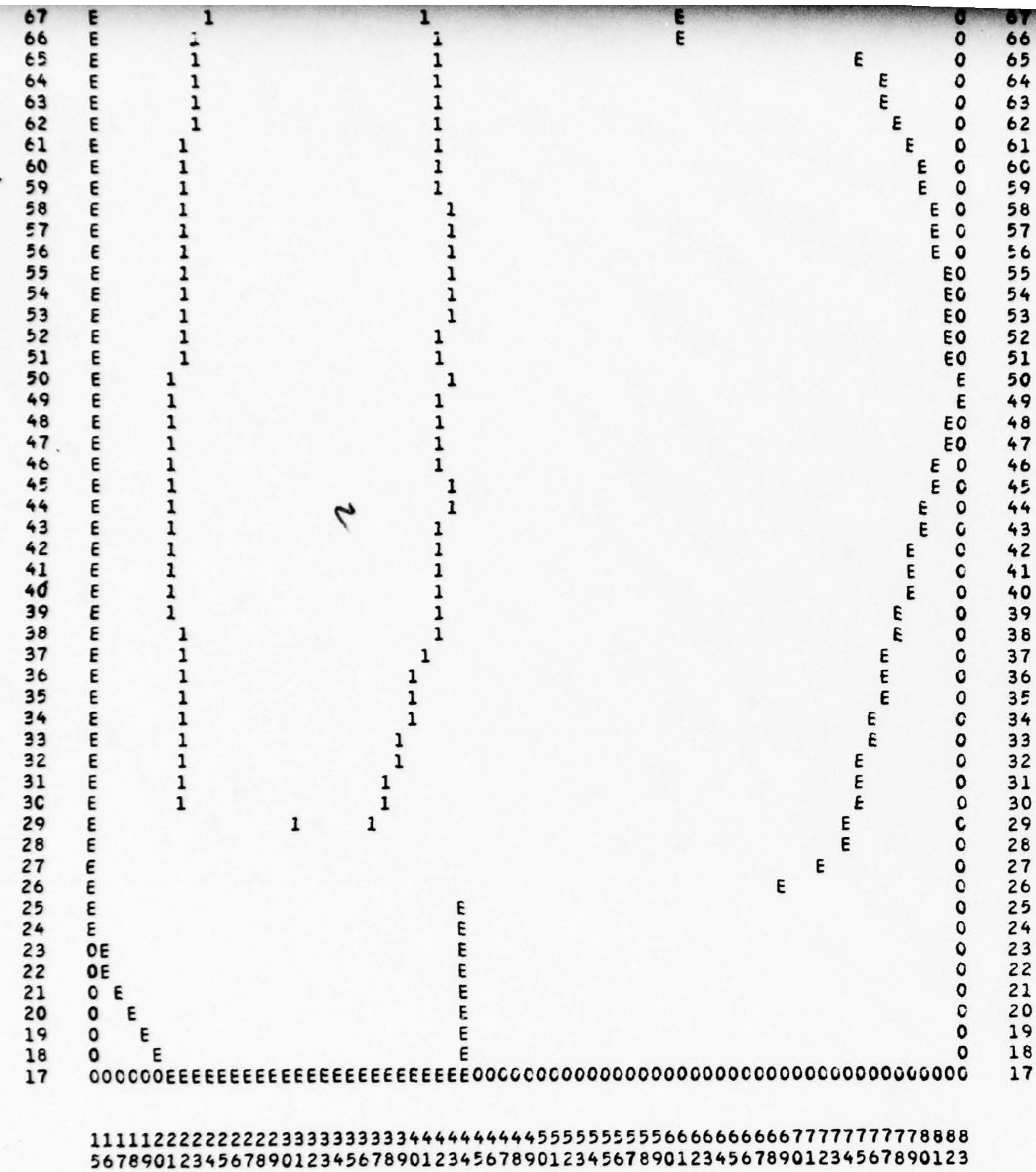
LETHAL AREA= 11.016 SQUARE METERS   AVERAGE PROBABILITY OF DAMAGE = .064 OVER 171.521 SQUARE METERS

OUTPUT OF THE PROBABILITY OF DAMAGE MATRIX HAS BEEN SUPPRESSED

THE PROBABILITY CONTOURS FOR THIS TARGET, HOB, AND DAMAGE CRITERION FOLLOW. THE NUMBERS ARE PD*10. (TRUNCATED), AND E=.0005, *=1.0

```
        111112222222222333333333344444444445555555555666666666677777777778888
        567890123456789012345678901234567890123456789012345678901234567890123

100     EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE000000000000000000000000000000000  100
 99     E                                  E                                0   99
 98     E                                 E                                 0   98
 97     E                                 E                                 0   97
 96     E                                 E                                 0   96
 95     E                                 E                                 0   95
 94     E                                 E                                 0   94
 93     E                                 E                                 0   93
 92     E                                 E                                 0   92
 91     E                                 E                                 0   91
 90     E                                 E                                 0   90
 89     E                                 E                                 0   89
 88     E                                E                                  0   88
 87     E                                E                                  0   87
 86     E                                E                                  0   86
 85     E                                E                                  0   85
 84     E                                E                                  0   84
 83     E                                E                                  0   83
 82     E                                E                                  0   82
 81     E                                E                                  0   81
 80     E                                E                                  0   80
 79     E                                E                                  0   79
 78     E                                E                                  0   78
 77     E                               E                                   0   77
 76     E                               E                                   0   76
 75     E                               E                                   0   75
 74     E                                              E                    0   74
 73     E                                              E                    0   73
 72     E                    11                        E                    0   72
 71     E         1            1                       E                    0   71
 70     E        1              1                      E                    0   70
 69     E        1               1                     E                    0   69
 68     E        1                1                    E                    0   68
 67     E        1                1                   E                     0   67
 66     E       1                  1                  E                     0   66
 65     E       1                  1                                E       0   65
 64     E       1                  1                                  E     0   64
 63     E       1                  1                                  E     0   63
 62     E       1                  1                                   E    0   62
 61     E      1                   1                                    E   0   61
 60     E      1                   1                                     E  0   60
```



THESE DATA ARE ON THE DAMAGE FUNCTION FILE (DFCN348) AS RECORD NUMBER 7

AVERAGE PD WITHIN EACH REGION	LETHAL AREA IN METERS**2		DAMAGE AREA IN METERS**2	
	PER REGION	CUMULATIVE	PER REGION	CUMULATIVE
.134	4.787	4.787	35.675	35.675
.046	6.229	11.016	135.846	171.521

AVERAGE PD WITHIN   RECTANGULARIZED, UNNORMALIZED RANGE DATA   RECTANGULARIZED, UNNORMALIZED DEFLECTION DATA

The following pa

AVERAGE PD WITHIN EACH REGION	LETHAL AREA IN METERS**2		DAMAGE AREA IN METERS**2	
	PER REGION	CUMULATIVE	PER REGION	CUMULATIVE
.134	4.787	4.787	35.675	35.675
.046	6.229	11.016	135.846	171.521

AVERAGE PD WITHIN EACH REGION	RECTANGULARIZED, UNNORMALIZED RANGE DATA				RECTANGULARIZED, UNNORMALIZED DEFLECTION DATA			
	EXTENT OF REGION		LOCATION OF REGION CENTER WRT PROJECTILE		EXTENT OF REGION		LOCATION OF REGION CENTER WRT PROJECTILE	
3	METERS	CELLS	METERS	CELLS	METERS	CELLS	METERS	CELLS
.134	4.674	23.0	-4.775	-23.5	8.941	44.0	10.160	50.0
.046	14.021	69.0	-1.321	-6.5	17.069	84.0	11.786	58.0

FIGURE C-9. SOME OUTPUT OF PROGRAM SECTION 1 FOR HOB = 12 METERS.

The following page is blank.

TABLE C-5. TABULATED RESULTS FROM PROGRAM SECTION I

Plate	HOB in Meters	Lethal Area in $Meters^2$	Damage Area in $Meters^2$	Average Probability of Damage
A	.3	13.626	13.626	1.
	5.	9.323	18.127	.514
	12.	8.930	62.431	.143
B	.3	20.232	20.232	1.
	5.	21.535	95.835	.225
	12.	10.910+	171.521+	.064$\pm$
F	.3	11.613	11.613	1.
	5. 12.	Not computed; known to be 0. from section IV output		
G	.3	0.	0.	0.
	5.	0.	0.	0.
	12.	0.	0.	0.

The following page is blank.

```
PROBABILITY OF SUSTAINING AT LEAST  1 PERFORATIONS          HOB =   .30 METERS          CELL = 4.00 INCHES

THE TARGET/HOB COMBINATION WAS ASSEMBLED FROM RECORDS    1,    3,    4, AND    2

                            THE CENTER X OFFSETS WERE    C,    2,   21, AND   40

                            THE CENTER Y OFFSETS WERE    0,    C,    0, AND    0

NXX=100 CELLS    NYY=100 CELLS    PROJECTILE LOCATION WAS (  55.5,     .5)

LETHAL AREA= 24.361 SQUARE METERS    AVERAGE PROBABILITY OF DAMAGE =1.000 OVER   24.361 SQUARE METERS


                                    OUTPUT OF THE PROBABILITY OF DAMAGE MATRIX HAS BEEN SUPPRESSED


THE PROBABILITY CONTOURS FOR THIS TARGET, HOB, AND DAMAGE CRITERION FOLLOW. THE NUMBERS ARE PD*10. (TRUNCATED), AND E=.0005, *=1.0

     00000001111111111222222222233333333334444444444555555555566666666667777777777888888888
     34567890123456789012345678901234567890123456789012345678901234567890123456789012345678

 28  0000000000000000**********************************************************************   28
 27  0               *                                                                    *   27
 26  0               *                                                                    *   26
 25  *                                                                                    *   25
 24  *                                                                                    *   24
 23  *                                                                                    *   23
 22  *                                                                                    *   22
 21  *                                                                                    *   21
 20  *                                                                                    *   20
 19  *                                                                                    *   19
 18  *                                                                                    *   18
 17  *                                                                                    *   17
 16  *                                                                                    *   16
 15  *                                                                                    *   15
 14  *                                                                                    *   14
 13  *                                                                                    *   13
 12  *                                                                                    *   12
 11  *                                                                                    *   11
 10  *                                                                                    *   10
  9  *                                                                                    *    9
  8  *                                                                                    *    8
  7  *                                                                                    *    7
  6  *                                                                                    *    6
  5  *                                                                                    *    5
  4  *                                                                                    *    4
  3  *                                                                                    *    3
  2  *                                                                                    *    2
  1  **************************************************************************************    1

     00000001111111111222222222233333333334444444444555555555566666666667777777777888888888
     34567890123456789012345678901234567890123456789012345678901234567890123456789012345678


THESE DATA ARE ON THE DAMAGE FUNCTION FILE (DFCN348    ) AS RECORD NUMBER    8


                           AVERAGE PD WITHIN      LETHAL AREA IN METERS**2      DAMAGE AREA IN METERS**2
                              EACH REGION
```

3456789012345678901234567890123456789012345678901234567890123456789012345678

THESE DATA ARE ON THE DAMAGE FUNCTION FILE (DFCN348) AS RECORD NUMBER 8

AVERAGE PD WITHIN EACH REGION	LETHAL AREA IN METERS**2		DAMAGE AREA IN METERS**2	
	PER REGION	CUMULATIVE	PER REGION	CUMULATIVE
1.00C	24.361	24.361	24.361	24.361

AVERAGE PD WITHIN EACH REGION	RECTANGULARIZED, UNNORMALIZED RANGE DATA				RECTANGULARIZED, UNNORMALIZED DEFLECTION DATA			
	EXTENT OF REGION		LOCATION OF REGION CENTER WRT PROJECTILE		EXTENT OF REGION		LOCATION OF REGION CENTER WRT PROJECTILE	
	METERS	CELLS	METERS	CELLS	METERS	CELLS	METERS	CELLS
1.000	8.738	86.0	-1.C16	-10.0	2.845	28.0	1.422	14.C

FIGURE C-10. OUTPUT OF PROGRAM SECTION II FOR HOB = .3 METERS.

The following page is blank.

PROBABILITY OF SUSTAINING AT LEAST 1 PERFORATIONS   HOB = 5.00 METERS   CELL = 8.00 INCHES

THE TARGET/HOB COMBINATION WAS ASSEMBLED FROM RECORDS 5, 9, AND 7

THE CENTER X OFFSETS WERE 0, 1, AND 21

THE CENTER Y OFFSETS WERE 0, 0, AND 0

NXX=100 CELLS   NYY=100 CELLS   PROJECTILE LOCATION WAS (55.5, .5)

LETHAL AREA= 30.522 SQUARE METERS   AVERAGE PROBABILITY OF DAMAGE = .289 OVER 105.538 SQUARE METERS

OUTPUT OF THE PROBABILITY OF DAMAGE MATRIX HAS BEEN SUPPRESSED

THE PROBABILITY CONTOURS FOR THIS TARGET, HOB, AND DAMAGE CRITERION FOLLOW. THE NUMBERS ARE PD*10. (TRUNCATED), AND E=.0005, *=1.0

```
     3333334444444444555555555566666666667777777
     4567890123456789012345678901234567890123456

76   0000EEEEEEEEEEEEEEEEEEEEEEEEEE0000C0000CC00   76
75   0 E                          E            0   75
74   E                            E            0   74
73   E                            E            0   73
72   E                            E            0   72
71   E                            E            0   71
70   E                            E            C   70
69   E                             E           0   69
68   E                             E           0   68
67   E                            E            0   67
66   E                            E            0   66
65   E                            E            C   65
64   E                            E            0   64
63   E                            E            C   63
62   E                            E            0   62
61   E                            E            C   61
60   E                            E            0   60
59   E                            E            0   59
58   E                           E             0   58
57   E     1               1     E             C   57
56   E    1                 1    E             0   56
55   E    1                 1    E             0   55
54   E    1                 1    E             0   54
53   E   1                  1    E             0   53
52   E   1                   1   E             0   52
51   E   1                  1    E             0   51
50   E   1                 1     E             0   50
49   E   1                 1     E             0   49
48   E  1                    1  E              0   48
47   E  1                    1  E              0   47
46   E  1                    1  E              0   46
45   E  1                     1      E         0   45
44   E 1  2                 2 1       E        0   44
43   E 1 2                  2 1       E        0   43
42   E 1 2                   2 1      E        0   42
41   E 1 2                   2 1       E       0   41
40   E 12      3           3 2 1      E        0   40
39   E 12     3            3 2 1      E        0   39
38   E1 2   3               3 21      E        0   38
37   E1 2   3               3 2 1            E 0   37
36   E1 2  3                3 2  1           E 0   36
35   E1 2 3             4  4 32    1          E0   35
34   E1 23        4         43 2    1         E0   34
```

```
39  E 12      3              3 2 1        E          0   39
38  E1 2    3                 3 21        E          0   38
37  E1 2    3                 3 2 1               E  0   37
36  E1 2   3                  3 2  1              E  0   36
35  E1 2 3               4   4 32     1             E0   35
34  E1 23          4          43 2     1            E0   34
33  E12 3        4            4 3 2      1          E0   33
32  E12 3       4              43    2     1         E   32
31  E12 3       4        5   5 4 3      2     1      E   31
30  E123      4      5        5 4 3     2     1      E   30
29  E123     4      5         5 4   3     2   1     E0   29
28  E123   4     5             54    3    2   1     E0   28
27  E123 4    5       6       654     3   2   1     E0   27
26  E13 4    5      6         65 4    3   2   1   E  0   26
25  E234    5     6           6 5 4   3   2   1   E  0   25
24  E234   5    6              65   4 3  2  1  E     0   24
23  E234 5   6        7       7 65   4 3  2  1  E    0   23
22  E234 5 6        7          765   4 3 2  1  E     0   22
21  E234 5   6      7         7 65   43  2  1 E      0   21
20  E24   5 6      7           765   4 3 2  1 E      0   20
19  E24   5 6     7            765   43  2 1 E       0   19
18  E24   5 6     7           7 65   43  2 1 E       0   18
17  E234 5   6     7          7 65   43  2 1E        0   17
16  E234   5   6               65   4 3  21 E        0   16
15  E234   5   6               65   4 3 2 1E         0   15
14  E234 5     6               65 4  3 2 1E          0   14
13  E234 5     6               654  3  21 E          0   13
12  E13 45       6             654 3   21E           0   12
11  E1234     5      6         643     2 1E          0   11
10  OE124   5      6           65 3     21E          0   10
 9  OE 24 5     6              654  3  21E           0    9
 8  0 E245   6    7            765 4   32 1          0    8
 7  C   25 6   7   8           876 5 4 321           0    7
 6  0   356 7    8               8  6 54321          0    6
 5  0   36 7   8     9         98 7 6 5431           0    5
 4  0   36 7 8     9           98   7 6 531          0    4
 3  0   467    8   9             9 8 7 6531          C    3
 2  0   47   8    9              9 8 7 6531          0    2
 1  000477788889999999999999999887766531000000000        1

    3333334444444444555555555566666666667777777
    4567890123456789012345678901234567890123456
```

THESE DATA ARE ON THE DAMAGE FUNCTION FILE (DFCN348   ) AS RECORD NUMBER   9

AVERAGE PD WITHIN EACH REGION	LETHAL AREA IN METERS**2		DAMAGE AREA IN METERS**2	
	PER REGION	CUMULATIVE	PER REGION	CUMULATIVE
.936	2.512	2.512	2.684	2.684
.848	1.856	4.367	2.188	4.872
.732	3.508	7.876	4.790	9.662
.656	6.393	14.268	9.745	19.407
.552	3.371	17.639	6.111	25.518
.450	3.341	20.981	7.432	32.950
.346	3.187	24.168	9.208	42.158
.251	2.457	26.624	9.786	51.943
.142	2.384	29.009	16.805	68.749
.041	1.514	30.522	36.790	105.538

AVERAGE PD WITHIN EACH REGION	RECTANGULARIZED, UNNORMALIZED RANGE DATA				RECTANGULARIZED, UNNORMALIZED DEFLECTION DATA			
	EXTENT OF REGION		LOCATION OF REGION CENTER WRT PROJECTILE		EXTENT OF REGION		LOCATION OF REGION CENTER WRT PROJECTILE	
	METERS	CELLS	METERS	CELLS	METERS	CELLS	METERS	CELLS

The following

.656	6.393	14.268	9.745	19.407
.552	3.371	17.639	6.111	25.518
.450	3.341	20.981	7.432	32.950
.346	3.187	24.168	9.208	42.158
.251	2.457	26.624	9.786	51.943
.142	2.384	29.009	16.805	68.749
.041	1.514	30.522	36.790	105.538

AVERAGE PD WITHIN EACH REGION	RECTANGULARIZED, UNNORMALIZED RANGE DATA				RECTANGULARIZED, UNNORMALIZED DEFLECTION DATA			
	EXTENT OF REGION		LOCATION OF REGION CENTER WRT PROJECTILE		EXTENT OF REGION		LOCATION OF REGION CENTER WRT PROJECTILE	
3	METERS	CELLS	METERS	CELLS	METERS	CELLS	METERS	CELLS
.936	2.845	14.0	-.813	-4.0	1.016	5.0	.508	2.5
.848	4.064	20.0	-1.016	-5.0	1.422	7.0	.711	3.5
.732	5.080	25.0	-1.118	-5.5	4.674	23.0	2.337	11.5
.656	5.486	27.0	-.914	-4.5	5.486	27.0	2.743	13.5
.552	5.690	28.0	-.813	-4.0	6.299	31.0	3.150	15.5
.450	6.096	30.0	-1.016	-5.0	7.112	35.0	3.556	17.5
.346	6.299	31.0	-.914	-4.5	8.128	40.0	4.064	20.0
.251	6.706	33.0	-.914	-4.5	7.925	39.0	4.978	24.5
.142	7.518	37.0	-.508	-2.5	11.582	57.0	5.791	28.5
.041	8.738	43.0	-.102	-.5	14.021	69.0	8.433	41.5

FIGURE C-11. OUTPUT OF PROGRAM SECTION II FOR HOB = 5 METERS.

The following page is blank.

PROBABILITY OF SUSTAINING AT LEAST 1 PERFORATIONS   HOB = 12.00 METERS   CELL = 8.00 INCHES

THE TARGET/HOB COMBINATION WAS ASSEMBLED FROM RECORDS   6,   10, AND   8

THE CENTER X OFFSETS WERE   0,   1, AND   21

THE CENTER Y OFFSETS WERE   0,   0, AND   0

NXX=100 CELLS   NYY=100 CELLS   PROJECTILE LOCATION WAS (55.5,   .5)

LETHAL AREA= 19.894 SQUARE METERS   AVERAGE PROBABILITY OF DAMAGE = .093 OVER 214.958 SQUARE METERS

OUTPUT OF THE PROBABILITY OF DAMAGE MATRIX HAS BEEN SUPPRESSED

THE PROBABILITY CONTOURS FOR THIS TARGET, HOB, AND DAMAGE CRITERION FOLLOW. THE NUMBERS ARE PD*10. (TRUNCATED), AND E=.0005, *=1.0

```
      111111222222222233333333334444444444555555555566666666667777777777888
      456789012345678901234567890123456789012345678901234567890123456789012

100   EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE000000000000000000000000000000000  100
 99   E                                  E                                0   99
 98   E                                 E                                 0   98
 97   E                                 E                                 0   97
 96   E                                 E                                 0   96
 95   E                                 E                                 0   95
 94   E                                 E                                 0   94
 93   E                                 E                                 0   93
 92   E                                 E                                 0   92
 91   E                                 E                                 0   91
 90   E                                 E                                 0   90
 89   E                                 E                                 0   89
 88   E                                E                                  0   88
 87   E                                E                                  0   87
 86   E                                E                                  0   86
 85   E                                E                                  0   85
 84   E                                E                                  0   84
 83   E                                E                                  0   83
 82   E                                E                                  0   82
 81   E                                E                                  0   81
 80   E                                E                                  0   80
 79   E                                E                                  0   79
 78   E                                E                                  0   78
 77   E                               E                                   0   77
 76   E                               E                                   0   76
 75   E                               E                                   0   75
 74   E                                              E                    0   74
 73   E                                              E                    0   73
 72   E                    11                        E                    0   72
 71   E         1            1                       E                    0   71
 70   E        1              1                      E                    0   70
 69   E        1               1                     E                    0   69
 68   E        1                1                    E                    0   68
 67   E        1                1                   E                     0   67
 66   E       1                  1                  E                     0   66
 65   E       1                  1                                E       0   65
 64   E       1                  1                                  E     0   64
 63   E       1                  1                                  E     0   63
 62   E       1                  1                                   E    0   62
 61   E      1                   1                                    E   0   61
 60   E      1                   1                                     E  0   60
 59   E      1                   1                                     E  0   59
 58   E      1                    1                                     E 0   58
 57   E      1                    1                                     E 0   57
 56   E      1                    1                                     E 0   56
```

```
  60    E       1                      1                                           E  0    60
  59    E       1                      1                                           E  C    59
  58    E       1                       1                                           E 0    58
  57    E       1                       1                                           E 0    57
  56    E       1                       1                                           E 0    56
  55    E       1                       1                                            EO    55
  54    E       1                       1                                            EO    54
  53    E       1                       1                                            EO    53
  52    E       1                      1                                             EO    52
  51    E       1                      1                                             EO    51
  50    E      1                        1                                             E    50
  49    E      1                       1                                              E    49
  48    E      1                       1                                             EC    48
  47    E      1                       1                                             E0    47
  46    E      1                       1                                            E C    46
  45    E      1                        1                                           E 0    45
  44    E      1                        1                                          E  0    44
  43    E      1                       1                                           E  0    43
  42    E      1                       1                                         E    0    42
  41    E      1                       1                                         E    0    41
  40    E      1                       1                                         E    0    40
  39    E      1                       1                                        E     0    39
  38    E       1                      1                                        E     0    38
  37    E       1             2       1                                        E      0    37
  36    E       1                    1                                         E      0    36
  35    E       1                    1                                         E      0    35
  34    E       1                    1                                        E       0    34
  33    E       1                   1                                         E       0    33
  32    E       1                   1                                        E        0    32
  31    E       1                   1                                        E        0    31
  30    E       1                   1                                        E        0    30
  29    E       1                    1                                      E         0    29
  28    E       1                    1                                      E         0    28
  27    E       1                     1                                  E            0    27
  26    E       1                     1                               E               0    26
  25    E      1                      1  E                                            0    25
  24    E       1                     1  E                                            0    24
  23    OE      1                     1  E                                            0    23
  22    CE     1                      1                                E              0    22
  21    O E    1                       1                                   E          0    21
  20    0  E   1                       1                                   E          0    20
  19    0  E    1                      1                                   E          0    19
  18    0  E   1              2   2    1                                   E          0    18
  17    0  E   1          2       2    1E       E1              1          E          0    17
  16    C  E   1  2                2   1E     E1                 1         E          C    16
  15    C  E   1  2                2    1E E1                     1        E          0    15
  14    0  E   1  2                 2    1E1                      1        E          0    14
  13    0  E   1  2                 2                             1      E            C    13
  12    0  E   1  2                  21         122               1      E            0    12
  11    0  E 1    2                   21     12    2              1      E            C    11
  10    0  E 1   2               33      2112           2         1      E            0    10
   9    0  E 1   2          3      3     212             2        1      E            0     9
   8    0  E 1   2         3         3                    2       1     E             0     8
   7    0  E 1   2        3           3                   2       1     E             0     7
   6    0  E 1   2       3             3                  2       1     E             0     6
   5    0  E 1   2       3              3                 2       1     E             0     5
   4    0  E 1   2       3              3                 2       1     E             0     4
   3    0  E 1   2      3               3                 2       1     E             0     3
   2    0  E 1   2      3               3                 2       1     E             0     2
   1    COOEE1112222222233333333333333333322222222222222222211111111EEEEEE00000000000000    1

        1111112222222222333333333344444444445555555555666666666677777777778888
        4567890123456789012345678901234567890123456789012345678901234567890123
```

THESE DATA ARE ON THE DAMAGE FUNCTION FILE (DFCN348   ) AS RECORD NUMBER 10

AVERAGE PD WITHIN EACH REGION	LETHAL AREA IN METERS**2 PER REGION	LETHAL AREA IN METERS**2 CUMULATIVE	DAMAGE AREA IN METERS**2 PER REGION	DAMAGE AREA IN METERS**2 CUMULATIVE
.335	1.591	1.591	4.748	4.748

THESE DATA ARE ON THE DAMAGE FUNCTION FILE (DFCN348) AS RECORD NUMBER 10

AVERAGE PD WITHIN EACH REGION	LETHAL AREA IN METERS**2 PER REGION	LETHAL AREA IN METERS**2 CUMULATIVE	DAMAGE AREA IN METERS**2 PER REGION	DAMAGE AREA IN METERS**2 CUMULATIVE
.335	1.591	1.591	4.748	4.748
.249	3.818	5.469	15.360	20.108
.141	8.121	13.530	57.435	77.544
.046	6.365	19.894	137.415	214.958

3

AVERAGE PD WITHIN EACH REGION	RECTANGULARIZED, UNNORMALIZED RANGE DATA				RECTANGULARIZED, UNNORMALIZED DEFLECTION DATA			
	EXTENT OF REGION		LOCATION OF REGION CENTER WRT PROJECTILE		EXTENT OF REGION		LOCATION OF REGION CENTER WRT PROJECTILE	
	METERS	CELLS	METERS	CELLS	METERS	CELLS	METERS	CELLS
.335	3.048	15.0	-4.166	-20.5	2.032	10.0	1.016	5.0
.249	7.518	37.0	-3.150	-15.5	3.658	18.0	1.829	9.0
.141	9.550	47.0	-2.743	-13.5	14.630	72.0	7.315	36.0
.046	14.021	69.0	-1.524	-7.5	20.320	100.0	10.160	50.0

FIGURE C-12. OUTPUT OF PROGRAM SECTION II FOR HOB = 12 METERS.

The following page is blank.

AN INDEX OF THE EXPECTED PERFORATIONS FILE (PMAT348) FOLLOWS

RECORD	NXX	NYY	CELL	HOB	IROW	ICOL	LYMYB	LYMYT	LYMXL	LYMXR	A	B	NSIDE	NX	NY	NZ	T	THCK
1	100	100	4.00	.30	0	55	1	20	23	86	0.0	0.0	0	42	16	0	78.0	1.50
2	100	100	4.00	.30	0	55	1	2	55	56	135.0	0.0	1	0	18	6	33.0	2.00
3	100	100	4.00	.30	0	55	1	28	21	90	0.0	41.0	2	46	16	8	78.0	1.60
4	100	100	4.00	.30	0	55	1	25	24	66	30.0	0.0	1	0	26	22	78.0	2.00
5	100	100	6.00	5.00	0	55	1	15	37	67	0.0	0.0	0	21	6	0	78.0	1.50
6	100	100	8.00	12.00	0	55	1	34	17	72	0.0	0.0	0	21	8	0	78.0	1.50
7	100	100	8.00	5.00	0	55	1	2	55	56	135.0	0.0	1	0	9	3	33.0	2.00
8	100	100	8.00	12.00	0	55	1	2	55	56	135.0	0.0	1	0	9	3	33.0	2.00
9	100	100	6.00	5.00	0	55	8	76	35	77	0.0	41.0	2	23	6	4	78.0	1.60
10	100	100	8.00	12.00	0	55	17	100	15	83	0.0	41.0	2	23	8	4	78.0	1.60

FIGURE C-13. INDEX OF THE EXPECTED PERFORATIONS MATRIX FILE.

Next page is blank.

A TABULATION OF THE DAMAGE FUNCTION FILE (DFCN348) FOLLOWS

RECORD NUMBER	POINT NUMBER	SYMMETRY CHARACTER	DAMAGE CRITERION	LYMXL	LYMXR	LYMYB	LYMYT	PROJECTILE LOCATION	HOB (METERS)	CELL (INCHES)	NUMBER OF REGIONS
8	1	C	1	3	68	1	28	(55.5, .5)	.30	4.00	1

DAMAGE FUNCTION REGIONS

PDBAR	LETHAL AREA	DAMAGE AREA	DIMENSIONS IN METERS (UNNORMALIZED) RANGE	DEFLECTION	OFFSET OF MIDPOINT WRT PROJECTILE IN METERS RANGE	DEFLECTION
1.000	8.738	2.845	-1.016	1.422	24.361	24.361

RECORD NUMBER	POINT NUMBER	SYMMETRY CHARACTER	DAMAGE CRITERION	LYMXL	LYMXR	LYMYB	LYMYT	PROJECTILE LOCATION	HOB (METERS)	CELL (INCHES)	NUMBER OF REGIONS
9	2	C	1	34	76	1	76	(55.5, .5)	5.00	8.00	10

DAMAGE FUNCTION REGIONS

PDBAR	LETHAL AREA	DAMAGE AREA	DIMENSIONS IN METERS (UNNORMALIZED) RANGE	DEFLECTION	OFFSET OF MIDPOINT WRT PROJECTILE IN METERS RANGE	DEFLECTION
.041	8.738	14.021	-.102	8.433	1.514	36.790
7.518	11.582	-.508	5.791	2.384	16.805	.251
7.925	-.914	4.978	2.457	9.786	.346	6.299
-.914	4.064	3.187	9.208	.450	6.096	7.112
3.556	3.341	7.432	.552	5.690	6.299	-.813
3.371	6.111	.656	5.486	5.486	-.914	2.743
9.745	.732	5.080	4.674	-1.118	2.337	3.506
.848	4.064	1.422	-1.016	.711	1.856	2.188
2.845	1.016	-.813	.508	2.512	2.684	.038
.024	.017	.010	.003	0.000	0.000	0.000

RECORD NUMBER	POINT NUMBER	SYMMETRY CHARACTER	DAMAGE CRITERION	LYMXL	LYMXR	LYMYB	LYMYT	PROJECTILE LOCATION	HOB (METERS)	CELL (INCHES)	NUMBER OF REGIONS
10	3	C	1	14	62	1	100	(55.5, .5)	12.00	8.00	4

DAMAGE FUNCTION REGIONS

PDBAR	LETHAL AREA	DAMAGE AREA	DIMENSIONS IN METERS (UNNORMALIZED)	OFFSET OF MIDPOINT WRT PROJECTILE IN METERS

The fol

10 3 0 1 14 62 1 100 (55.5, .5) 12.00 8.00 4

DAMAGE FUNCTION REGIONS

PDBAR	LETHAL AREA	DAMAGE AREA	DIMENSIONS IN METERS (UNNORMALIZED) RANGE	DEFLECTION	OFFSET OF MIDPOINT WRT PROJECTILE IN METERS RANGE	DEFLECTION
.046	14.021	20.320	-1.524	10.160	6.365	137.415
9.550	14.630	-2.743	7.315	8.121	57.435	.249
3.658	-3.150	1.829	3.818	15.360	.335	3.048
-4.166	1.016	1.591	4.746	.450	6.096	7.112

2

FIGURE C-14. A TABULATION OF THE DAMAGE FUNCTION FILE. RECORDS NUMBERED 1 THRU 7 HAVE BEEN DELETED BY THE AUTHOR.

The following page is blank.

results would have shown that plate G could have been eliminated from the section II deck, since it contributed nothing. Although very little would have been gained in this example, in some cases this sort of "one-step-at-a-time" analysis can save significant amounts of machine time.

We have already obtained some results of value for our APC model: lethal areas, average damage probabilities, probability of vulnerability of a single APC. Now we will use program section III to obtain measures of the damage sustained by the APC force fired upon by the howitzer battery (Figure C-1). Figures C-10 through C-12 provide contour plots for the damage functions, and we see that in each case, the damage function is "fully nested" (see paragraph 6.1). The damage functions input to program section III are constructed from those output by program section II using judgement and experience.

In this example, we make two different approximations to the damage functions output by program section II in order to give some rough idea of how results for program section III might depend upon the input damage function. The input data deck is at Figure C-15. The first approximation results in "cookie cutter" damage functions; that is, all the different probability of damage regions are combined to obtain one region with a constant probability of damage. For the second approximation, we rectangularize (maintaining normalization) the various probability of damage regions, without combining any of them.

Results of program section III for these approximations to the damage functions are at Figures C-16 and C-17. One should not assume that the differences in results for different approximations to the damage functions will always be as small as they are in this example.

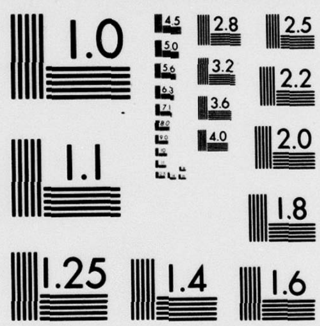
```
ENCOUNTER 6.        2.         8.         3.         12.
-20.      20.       0.         0.         20.        -20.
100.      60.       20.        -20.       -60.       -100.
0.        0.
75.       25.
1.        2.        3.         4.         5.         10.        15.        20.
36.       6.        45.        27.        .95
DFCN348   SYMMETRY  8.    1.
DFCN348   SYMMETRY  9.    10.
DFCN348   SYMMETRY  10.   4.
HOB DISTRIBUTION    3.         SYMMETRIC ARRAY       OPTIMIZE  5.
.3        5.        12.
.03       .80       .17
ENCOUNTER 6.        2.         8.         3.         12.
SAME CANNON
SAME TARGETS
SAME VOLLEYS
SAME ERRORS
DFCN348   SYMMETRY  8.
DFCN348   SYMMETRY  9.
DFCN348   SYMMETRY  10.
SAME HOB DISTRIBUTION          SYMMETRIC ARRAY       OPTIMIZE  5.
END OF DATA
```

FIGURE C-15. INPUT DATA DECK FOR PROGRAM SECTION III

AD-A068 976 ARMY MATERIEL SYSTEMS ANALYSIS ACTIVITY ABERDEEN PROV--ETC F/G 19/3
LIGHTLY ARMORED STRUCTURE VULNERABILITY ESTIMATION METHODOLOGY --ETC(U)
JAN 79 C J LAPOINTE
UNCLASSIFIED AMSAA-TR-254 NL

3 OF 3
AD
A068976

END
DATE
FILMED
7 — 79
DDC



MICROCOPY RESOLUTION TEST CHART

DAMAGE FUNCTION NUMBER 8 FROM FILE DFCN348 FOR HOB = .30 HAS 1 REGIONS, OF WHICH 0 ARE UNNESTED

PD	REGION DIMENSIONS		OFFSET OF MIDPOINT FROM TARGET CENTER	
	RANGE	DEFLECTION	RANGE	DEFLECTION
1.000	8.65	5.63	-1.02	0.00

DAMAGE FUNCTION NUMBER 9 FROM FILE DFCN348 FOR HOB = 5.00 HAS 1 REGIONS, OF WHICH 0 ARE UNNESTED

PD	REGION DIMENSIONS		OFFSET OF MIDPOINT FROM TARGET CENTER	
	RANGE	DEFLECTION	RANGE	DEFLECTION
.289	7.73	27.32	-.10	0.00

DAMAGE FUNCTION NUMBER 10 FROM FILE DFCN348 FOR HOB = 12.00 HAS 1 REGIONS, OF WHICH 0 ARE UNNESTED

PD	REGION DIMENSIONS		OFFSET OF MIDPOINT FROM TARGET CENTER	
	RANGE	DEFLECTION	RANGE	DEFLECTION
.093	12.18	35.30	-1.52	0.00

TARGET NUMBER 1 LOCATED AT (0.00, 75.00)

VOLLEY SIZE =	1	2	3	4	5	10	15	20
PD(HOB= .30) =	.007267	.014403	.021410	.028293	.035053	.067108	.096481	.123453
PD(HOB= 5.00) =	.009055	.017983	.026786	.035466	.044024	.085072	.123379	.159161
PD(HOB= 12.00) =	.005862	.011678	.017446	.023168	.028843	.056547	.083160	.106731

TARGET NUMBER 2 LOCATED AT (0.00, 25.00)

VOLLEY SIZE =	1	2	3	4	5	10	15	20
PD(HOB= .30) =	.007841	.015540	.023098	.030519	.037807	.072347	.103975	.133005
PD(HOB= 5.00) =	.009331	.019522	.029076	.038495	.047781	.092297	.133808	.172552
PD(HOB= 12.00) =	.006380	.012708	.018983	.025207	.031380	.061489	.090384	.118119

PD(HOB= 5.00) =	.009831	.019522	.029076	.038495	.047781	.092297	.133808	.172552
PD(HOB= 12.00) =	.006380	.012768	.018983	.025207	.031380	.061489	.090384	.118119

THE HOB DISTRIBUTION FOLLOWS

HOB =	.30	5.00	12.00
FRACTION OF POPULATION =	.030	.800	.170

PROBABILITIES OF DAMAGE AVERAGED OVER THE HOB DISTRIBUTION FOR EACH TARGET FOLLOW

VOLLEY SIZE =	1	2	3	4	5	10	15	20
TARGET NUMBER 1 =	.008459	.016804	.025037	.033160	.041174	.079684	.115735	.149516
TARGET NUMBER 2 =	.009184	.018244	.027181	.035997	.044694	.086461	.125531	.162112

THE SYSTEM (INCLUDING TARGET REFLECTIONS DUE TO SYMMETRY) DAMAGE PROBABILITIES FOLLOW

VOLLEY SIZE =	1	2	3	4	5	10	15	20
ALL SURVIVE =	.96518	.93172	.89958	.86669	.83900	.70685	.59793	.50781
EXACTLY ONE DAMAGED =	.03436	.06648	.09647	.12446	.15056	.25620	.32819	.37505
EXACTLY TWO DAMAGED =	.00046	.00178	.00388	.00668	.01013	.03480	.06750	.10379
AT LEAST ONE DAMAGED =	.03482	.06828	.10042	.13131	.16100	.29315	.40207	.49219
AT LEAST TWO DAMAGED =	.00046	.00180	.00395	.00684	.01043	.03695	.07388	.11714
AT LEAST THREE DAMAGED =	.00000	.00002	.00007	.00016	.00031	.00215	.00638	.01334
LESS THAN TWO DAMAGED =	.99954	.99820	.99605	.99316	.98957	.96305	.92612	.88286
LESS THAN THREE DAMAGED =	1.00000	.99998	.99993	.99984	.99969	.99785	.99362	.98666
AVERAGE PER TARGET =	.00882	.01752	.02611	.03458	.04293	.08307	.12063	.15581

PD'S FOR EACH TARGET WITH OPTIMUM FUZING (HOB= 5.00 METERS, ATTACK LEVEL= 5 ROUNDS PER CANNON)

VOLLEY SIZE =	1	2	3	4	5	10	15	20
TARGET NUMBER 1 =	.009055	.017983	.026786	.035466	.044024	.085072	.123379	.159161
TARGET NUMBER 2 =	.009831	.019522	.029076	.038495	.047781	.092297	.133808	.172552

THE SYSTEM (INCLUDING TARGET REFLECTIONS DUE TO SYMMETRY) DAMAGE PROBABILITIES FOLLOW

VOLLEY SIZE =	1	2	3	4	5	10	15	20

The followi

VOLLEY SIZE =	1	2	3	4	5	10	15	20
TARGET NUMBER 1 =	.009055	.017983	.026786	.035466	.044024	.085072	.123379	.159161
TARGET NUMBER 2 =	.0C9831	.019522	.029076	.038495	.047781	.092297	.133808	.172552

THE SYSTEM (INCLUDING TARGET REFLECTIONS DUE TO SYMMETRY) DAMAGE PROBABILITIES FOLLOW

VOLLEY SIZE =	1	2	3	4	5	10	15	20
ALL SURVIVE =	.96276	.92707	.89287	.86008	.82864	.68970	.57657	.48407
EXACTLY ONE DAMAGED =	.03671	.07087	.10263	.13212	.15948	.26852	.34043	.38515
EXACTLY TWO DAMAGED =	.00052	.00203	.00442	.00761	.01150	.03918	.07532	.11483
AT LEAST ONE DAMAGED =	.03724	.07293	.10713	.13992	.17136	.31030	.42343	.51593
AT LEAST TWO DAMAGED =	.00053	.00206	.00451	.00780	.01188	.04178	.08300	.13078
AT LEAST THREE DAMAGED =	.00000	.00003	.00009	.00020	.00037	.00260	.00767	.01596
LESS THAN TWO DAMAGED =	.99947	.99794	.99549	.99220	.98812	.95822	.91700	.86922
LESS THAN THREE DAMAGED =	1.00000	.99997	.99991	.99980	.99963	.99740	.99233	.98404
AVERAGE PER TARGET =	.00944	.01875	.02793	.03698	.04590	.08868	.12859	.16586

FIGURE C-16. RESULTS OF PROGRAM SECTION III FOR THE COOKIE CUTTER APPROXIMATIONS TO THE DAMAGE FUNCTIONS.

The following page is blank.

DAMAGE FUNCTION NUMBER 8 FROM FILE DFCN348 FOR HOB = .30 HAS 1 REGIONS, OF WHICH 0 ARE UNNESTED

PD	REGION DIMENSIONS RANGE	REGION DIMENSIONS DEFLECTION	OFFSET OF MIDPOINT FROM TARGET CENTER RANGE	OFFSET OF MIDPOINT FROM TARGET CENTER DEFLECTION
1.000	8.65	5.63	-1.02	0.00

DAMAGE FUNCTION NUMBER 9 FROM FILE DFCN348 FOR HOB = 5.00 HAS 10 REGIONS, OF WHICH 0 ARE UNNESTED

PD	REGION DIMENSIONS RANGE	REGION DIMENSIONS DEFLECTION	OFFSET OF MIDPOINT FROM TARGET CENTER RANGE	OFFSET OF MIDPOINT FROM TARGET CENTER DEFLECTION
.936	2.74	1.96	-.81	0.00
.848	3.73	2.61	-1.02	0.00
.732	3.73	5.18	-1.02	0.00
.656	4.41	8.81	-.91	0.00
.552	4.80	10.63	-.81	0.00
.450	5.31	12.40	-1.02	0.00
.346	5.72	14.75	-.91	0.00
.251	6.63	15.67	-.91	0.00
.142	6.68	20.58	-.89	0.00
.041	8.11	26.03	-.17	0.00

DAMAGE FUNCTION NUMBER 10 FROM FILE DFCN348 FOR HOB = 12.00 HAS 4 REGIONS, OF WHICH 0 ARE UNNESTED

PD	REGION DIMENSIONS RANGE	REGION DIMENSIONS DEFLECTION	OFFSET OF MIDPOINT FROM TARGET CENTER RANGE	OFFSET OF MIDPOINT FROM TARGET CENTER DEFLECTION
.335	2.67	3.56	-4.17	0.00
.249	6.43	6.26	-3.15	0.00
.141	7.11	21.80	-2.81	0.00
.046	12.18	35.30	-1.52	0.00

TARGET NUMBER 1 LOCATED AT (0.00, 75.00)

VOLLEY SIZE =	1	2	3	4	5	10	15	20
PD(HOB= .30) =	.007267	.014403	.021410	.028293	.035053	.067108	.096481	.123453
PD(HOB= 5.00) =	.009017	.017874	.026574	.035120	.043515	.063352	.119896	.153487
PD(HOB= 12.00) =	.005866	.011679	.017439	.023147	.028803	.056324	.082632	.107792

TARGET NUMBER 2 LOCATED AT (0.00, 25.00)

VOLLEY SIZE =	1	2	3	4	5	10	15	20
PD(HOB= .30) =	.007841	.015540	.023098	.030519	.037807	.072347	.103975	.133005

VOLLEY SIZE =	1	2	3	4	5	10	15	20
PD(HOB= .30) =	.007841	.015540	.023098	.030519	.037807	.072347	.103975	.133005
PD(HOB= 5.00) =	.009751	.019328	.028733	.037971	.047045	.090087	.129545	.165792
PD(HOB= 12.00) =	.006360	.012662	.018905	.025092	.031221	.061036	.089521	.116745

THE HOB DISTRIBUTION FOLLOWS

HOB =	.30	5.00	12.00
FRACTION OF POPULATION =	.030	.800	.170

PROBABILITIES OF DAMAGE AVERAGED OVER THE HOB DISTRIBUTION FOR EACH TARGET FOLLOW

VOLLEY SIZE =	1	2	3	4	5	10	15	20
TARGET NUMBER 1 =	.008429	.016717	.024866	.032879	.040760	.078270	.112859	.144817
TARGET NUMBER 2 =	.009118	.018081	.026893	.035558	.044078	.084616	.121974	.156470

2

THE SYSTEM (INCLUDING TARGET REFLECTIONS DUE TO SYMMETRY) DAMAGE PROBABILITIES FOLLOW

VOLLEY SIZE =	1	2	3	4	5	10	15	20
ALL SURVIVE =	.96537	.93220	.90043	.86999	.84081	.71189	.60674	.52038
EXACTLY ONE DAMAGED =	.03418	.06603	.09569	.12331	.14900	.25251	.32295	.36930
EXACTLY TWO DAMAGED =	.00045	.00175	.00381	.00655	.00990	.03357	.06442	.09821
AT LEAST ONE DAMAGED =	.03463	.06780	.09957	.13001	.15919	.28811	.39326	.47962
AT LEAST TWO DAMAGED =	.00046	.00177	.00388	.00671	.01019	.03559	.07032	.11033
AT LEAST THREE DAMAGED =	.00000	.00002	.00007	.00016	.00030	.00203	.00590	.01211
LESS THAN TWO DAMAGED =	.99954	.99823	.99612	.99329	.98981	.96441	.92968	.88967
LESS THAN THREE DAMAGED =	1.00000	.99998	.99993	.99984	.99970	.99797	.99410	.98789
AVERAGE PER TARGET =	.00877	.01740	.02588	.03422	.04242	.08144	.11742	.15064

PD'S FOR EACH TARGET WITH OPTIMUM FUZING (HOB= 5.00 METERS, ATTACK LEVEL= 5 ROUNDS PER CANNON)

VOLLEY SIZE =	1	2	3	4	5	10	15	20
TARGET NUMBER 1 =	.009017	.017874	.026574	.035120	.043515	.083352	.119896	.153487
TARGET NUMBER 2 =	.009751	.019328	.028733	.037971	.047045	.090087	.129545	.165792

THE SYSTEM (INCLUDING TARGET REFLECTIONS DUE TO SYMMETRY) DAMAGE PROBABILITIES FOLLOW

VOLLEY SIZE =	1	2	3	4	5	10	15	20
ALL SURVIVE =	.96299	.92765	.89389	.86164	.83081	.69567	.58690	.49867

The fo

VOLLEY SIZE =	1	2	3	4	5	10	15	20
TARGET NUMBER 1 =	.009017	.017874	.026574	.035120	.043515	.083352	.119896	.153487
TARGET NUMBER 2 =	.009751	.019328	.028733	.037971	.047045	.090087	.129545	.165792

THE SYSTEM (INCLUDING TARGET REFLECTIONS DUE TO SYMMETRY) DAMAGE PROBABILITIES FOLLOW

3

VOLLEY SIZE =	1	2	3	4	5	10	15	20
ALL SURVIVE =	.96299	.92765	.89389	.86164	.83081	.69567	.58690	.49867
EXACTLY ONE DAMAGED =	.03649	.07033	.10169	.13074	.15763	.26427	.33459	.37905
EXACTLY TWO DAMAGED =	.00052	.00200	.00434	.00744	.01121	.03762	.07149	.10797
AT LEAST ONE DAMAGED =	.03701	.07235	.10611	.13836	.16919	.30433	.41310	.50133
AT LEAST TWO DAMAGED =	.00052	.00202	.00442	.00762	.01157	.04006	.07851	.12228
AT LEAST THREE DAMAGED =	.00000	.00003	.00008	.00019	.00036	.00244	.00702	.01431
LESS THAN TWO DAMAGED =	.99948	.99798	.99558	.99238	.98843	.95994	.92149	.87772
LESS THAN THREE DAMAGED =	1.00000	.99997	.99992	.99981	.99964	.99756	.99298	.98569
AVERAGE PER TARGET =	.00938	.01860	.02765	.03655	.04528	.08672	.12472	.15964

FIGURE C-17. RESULTS OF PROGRAM SECTION III FOR THE APPROXIMATIONS OF GREATEST DETAIL TO THE DAMAGE FUNCTIONS.

The following page is blank.

APPENDIX D

CONTROL DECK FOR THE EXAMPLE OF APPENDIX C

This appendix presents and explains the control deck used to execute the program in the example of Appendix C. Detailed explanations of the commands and their parameters may be found in CDC system literature.

Figure D-1 presents the control deck used to obtain the first set of results, the program section IV output. Card 3 attaches the permanent file on which the program is maintained in relocatable form. Cards 4 and 5 make accessible the permanent file on which the data deck of Figures C-3 and C-4 is maintained. These data could be read in on cards, but they are maintained as a file for convenience. Card 7 causes the program to be loaded and executed with file S4 replacing the system input file.

Figure D-2 presents the control deck used to obtain the second set of results, the output of program sections I and II.

Cards 4 and 5 make accessible the permanent file on which the data deck of Figures C-3 and C-6 is maintained. Again, cards could be used. Card 9 causes the program to be loaded and executed with files S12, PMAT348, and DFCN348 replacing the system input, perforation matrix, and damage function files, respectively. Card 10 causes the perforation matrix file to be catalogued (made permanent) as file PMAT348 with a password whose future absence will prevent all access to the file except for reading it. Card 11 performs the same function for the damage function file, under the name DFCN348. If an additional run of program sections I and/or II were desired, with additional perforation matrices and damage functions being written on these same files, appropriate ATTACH or GETPF commands would be required between cards 7 and 9.

Figure D-3 presents the control deck used to obtain the third set of results, the output of program section III.

```
COLUMN              11111111112222222222333333333344444444445555555555666666666677777777778
NUMBER     12345678901234567890123456789012345678901234567890123456789012345678901234567890
 CARD
NUMBER
   1       JOBNAME,STMFZ.
   2       ACCOUNT,XXXXX.
   3       ATTACH,LASREL,LASVEMRELOCATABLE,ID=LAPOINTE.
   4       FILE,S4,FL=100,RT=Z,BT=C.
   5       GETPF,S4,SECTION4DATA,ID=LAPOINTE,ST=MFA.
   6       MAP,OFF.
   7       LASREL,S4.
   8       RETURN,S4,LASREL.
   9       789 (MULTIPUNCHED IN COLUMN 1)
  10       6789 (MULTIPUNCHED IN COLUMN 1)
```

FIGURE D-1. CONTROL DECK FOR PROGRAM SECTION IV

```
COLUMN             1111111111222222222233333333334444444444555555555566666666667777777778
NUMBER    12345678901234567890123456789012345678901234567890123456789012345678901234567890
 CARD
NUMBER
   1      JOBNAME,STMFZ.
   2      ACCOUNT,XXXXX.
   3      ATTACH,LASREL,LASVEMRELOCATABLE,ID=LAPOINTE.
   4      FILE,S12,FL=100,RT=Z,BT=C.
   5      GETPF,S12,SECTIONS1AND2DATA,ID=LAPOINTE,ST=MFA.
   6      REQUEST,PMAT348,*PF.
   7      REQUEST,DFCN348,*PF.
   8      MAP,OFF.
   9      LASREL,S12,,PMAT348,DFCN348.
  10      CATALOG,PMAT346,PMAT346,ID=LAPOINTE,XR=XXXXXXXXXX.
  11      CATALOG,DFCN348,DFCN348,ID=LAPOINTE,XR=XXXXXXXXXX.
  12      RETURN,S12,PMAT348,DFCN348,LASREL.
  13      789 (MULTIPUNCHED IN COLUMN 1)
  14      6789 (MULTIPUNCHED IN COLUMN 1)
```

FIGURE D-2. CONTROL DECK FOR PROGRAM SECTIONS I AND II

```
COLUMN            111111111122222222223333333333444444444455555555556666666666777777777 8
NUMBER   12345678901234567890123456789012345678901234567890123456789012345678901234567890
 CARD
NUMBER
   1     JOBNAME,STMFZ.
   2     ACCOUNT,XXXXX.
   3     ATTACH,LASREL,LASVEMRELOCATABLE,ID=LAPOINTE.
   4     FILE,S3,FL=100,RT=Z,BT=C.
   5     GETPF,S3,SECTION3DATA,ID=LAPOINTE,ST=MFA.
   6     ATTACH,PMAT348,PMAT348,ID=LAPOINTE.
   7     ATTACH,DFCN348,DFCN348,ID=LAPOINTE.
   8     MAP,OFF.
   9     LASREL,S3,,PMAT348,DFCN348.
  10     RETURN,S3,PMAT348,DFCN348,LASREL.
  11     789 (MULTIPUNCHED IN COLUMN 1)
  12     6789 (MULTIPUNCHED IN COLUMN 1)
```

FIGURE D-3. CONTROL DECK FOR PROGRAM SECTION III

ADDENDUM

Paragraph 4.1.1, page 8 , addresses the use of different linear combinations of the perforations sustained assuming:

a. All fragments within a spray leave the source point with that spray's median velocity.

b. All fragments within a spray leave the source point with that spray's maximum velocity.

Subsequent to the completion of this report, the program code was modified to allow the linear combination to be varied routinely, as input data. The coding which effects this is shown in Figure ADD-1. The sequence numbers (columns 82-87) indicate placement of the cards within the program shown in Appendix A.

The variable linear combination coefficients are F(1) and F(2). F(1) pertains to the perforations obtained with median velocities and F(2), to those obtained with maximum velocities. F(1) and F(2) are read in as real numbers with explicit decimal points on card number 1 of the BASIC DATA subdeck (see Figure 4-9), F(1) in columns 71 thru 75 and F(2) in columns 76 thru 80.

```
      COMMON /LINCOM/ F(2),P(2)                                         LASVM     C00203
C                                                                       LASVM     C00206
      DECODE(70,1115,AAA(2))FFILES,CELL1,FNXX,FNYY,FICOL,FIROW,F(1),F(2)LASV197*  C02060
     1  SCALE2,F(1),F(2)                                                LASV404W  004130
 1115 FORMAT(6E10.4,2F5.1)                                              LASVM     0C7935
     4 TO INPUT PLATE = 1./,F5.3,10X,34HLINEAR COMBINATION COEFFICIENTS LASV798   008070
     5= ,F5.3,1H,,F6.3/)                                                LASVM     008075
      COMMON /LINCOM/ FF(2),P(2)                                        GEOMT     010613
C                                                                       GEOMT     C10616
   30 PERF(II,JJ)=FF(1)*P(1)+FF(2)*P(2)                                 GEOMT59   011100
  110 PRFK=PRFK+FF(1)*P(1)+FF(2)*P(2)                                   GEOM141   011920
  120 PERF(II,JJ)=PRFK                                                  GEOM142   011940
  270 PRFK=PRFK+FF(1)*P(1)+FF(2)*P(2)                                   GEOM255   C13070
  280 PERF(II,JJ)=PRFK                                                  GEOM256   C13080
  420 PRFK=PRFK+FF(1)*P(1)+FF(2)*P(2)                                   GEOM374   014260
  430 PERF(II,JJ)=PRFK                                                  GEOM375   C14270
      PRFK=PRFK+FF(1)*P(1)+FF(2)*P(2)                                   GEOM413   014650
  480 PRFK=PRFK+FF(1)*P(1)+FF(2)*P(2)                                   GEOM422   014740
  490 PERF(II,JJ)=PRFK                                                  GEOM423   014750
  510 PRFK=PRFK+FF(1)*P(1)+FF(2)*P(2)                                   GEOM448   015000
  520 PERF(II,JJ)=PRFK                                                  GEOM449   015010
  660 PRFK=PRFK+FF(1)*P(1)+FF(2)*P(2)                                   GEOM564   016160
  670 PERF(II,JJ)=PRFK                                                  GEOM565   016170
      PRFK=PRFK+FF(1)*P(1)+FF(2)*P(2)                                   GEOM604   016560
  720 PRFK=PRFK+FF(1)*P(1)+FF(2)*P(2)                                   GEOM613   016650
  730 PERF(II,JJ)=PRFK                                                  GEOM614   016660
  750 PRFK=PRFK+FF(1)*P(1)+FF(2)*P(2)                                   GEOM639   016910
  760 PERF(II,JJ)=PRFK                                                  GEOM640   016920
      COMMON /LINCOM/ F(2),P(2)                                         DAMGE     018383
C                                                                       DAMGE     C18386
      P(1)=0.                                                           DAMGE28   018610
      P(2)=0.                                                           DAMGE     018615
      I=1                                                               DAMGE     018743
      IF(K3.GT.48)I=2                                                   DAMGE     018746
      P(I)=P(I)+SGDIN(K3)*STER*Q(K1,K2)                                 DAMGE42   C18750
```

FIGURE ADD-1. PROGRAM MODIFICATION.

DISTRIBUTION LIST

No. of Copies	Organization	No. of Copies	Organization
12	Commander Defense Documentation Center ATTN: TCA Cameron Station Alexandria, VA 22314	1	Commander US Army Missile R&D Command ATTN: DRDMI-C Redstone Arsenal, AL 35809
1	Commander US Army Materiel Development and Readiness Command ATTN: DRCBSI-L 5001 Eisenhower Avenue Alexandria, VA 22333	2	Commander US Army Tank-Automotive R&D Command ATTN: DRDTA-UL (Tech Lib) DRDTA-V Warren, MI 48090
1	Commander US Army Materiel Development and Readiness Command ATTN: DRCPA-S 5001 Eisenhower Avenue Alexandria, VA 22333	1	Commander US Army Tank Automotive R&D Command ATTN: DRDTA-RCKA (Mr. Lim) Warren, MI 48090
1	Commander US Army Materiel Development and Readiness Command ATTN: DRCDE-D 5001 Eisenhower Avenue Alexandria, VA 22333	1	Project Manager XM1 Tank System ATTN: CRCPM-GCM-SA (Mr. Dean) MAMP Warren, MI 48090
1	Commander US Army Armament Materiel Readiness Command ATTN: DRSAR-SA Rock Island, IL 61299	1	Commander US Army Mobility Equipment R&D Command ATTN: DRDME-O Fort Belvoir, VA 22060
1	Commander Rock Island Arsenal ATTN: Tech Lib Rock Island, IL 61299	2	Director US Army TRADOC Systems Analysis Activity ATTN: ATAA-SL ATAA-T White Sands Missile Range, NM 88002
1	Commander US Army Materiel Development and Readiness Command ATTN: DRCDE-R 5001 Eisenhower Avenue Alexandria, VA 22333	2	Chief Defense Logistics Studies Information Exchange US Army Logistics Management Center ATTN: DRXMC-D Fort Lee, VA 23801

DISTRIBUTION LIST (Cont)

No. of Copies	Organization
1	Commander US Army Electronics R&D Command ATTN: DRDEL-SA Fort Monmouth, NJ 07703
1	Commander US Army Concepts Analysis Agency 8120 Woodmont Avenue Bethesda, MD 20014
1	Commander US Army Armor Center and Fort Knox ATTN: ATZK-CD-MS (LTC Walters) Fort Knox, KY 40121

Aberdeen Proving Ground

Cdr, APG, ATTN: STEAP-MT-A (Mr. Martin)

Dir, BRL, ATTN: DRDAR-BLB (Mr. Johnson)

Dir, BRL, ATTN: STINFO Branch, Bldg 305

Dir, BRL, Bldg. 328

Dir, HEL, Bldg. 520

Cdr, TECOM, ATTN: DRSTE-CM-R (LTC Bagelonas)
DRSTE
DRSTE-CS-A